Genetic Algorithm Based Scheduling To Reduce Energy Consumption In Cloud

Thesis submitted in partial fulfillment of the requirements for the award of the degree of

Master of Engineering

in

Computer Science and Engineering

Submitted by

Paridhi Naithani
(Roll no: 801532038)

Under the supervision of

Dr. Rajesh Kumar

Professor

Computer Science and Engineering Department

Thapar University
Patiala-147004, Punjab, India

July 2017

Certificate

I hereby certify that the work which is being presented in the thesis entitled, ***Genetic Algorithm Based Scheduling To Reduce Energy Consumption In Cloud***, in partial fulfillment of the requirements for the award of degree of Master of Engineering in ***Computer Science and Engineering*** submitted in Computer Science and Engineering Department of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of Dr. Rajesh Kumar and refers other researchers work which are duly listed in the reference section.

The matter presented in the thesis has not been submitted for award of any other degree of this or any other University.

Paridhi

Paridhi Naithani

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

Dr. Rajesh Kumar
Professor,CSED

Acknowledgements

First, I would like to express my deep gratitude to my supervisor **Dr. Rajesh Kumar** for their invaluable advice and encouragement at every step of my PhD program. Without their unfailing support and belief in me, this thesis would not have been possible.

I am also thankful to **Prof. Prakash Gopalan**, Director of Thapar University, Patiala for providing me all the resources required. I express my deep sense of gratitude towards **Dr. Maninder Singh**, Head, Computer Science and Engineering Department of Thapar University, Patiala who has been constant source of motivation throughout the work.

I would also like to say thanks to all my friends for their support. I want to express my thankfulness to every person who contributed with either inspiring me or helping me in actual work to this thesis.

Finally, I would like to express my sincere and deep gratitude to my parents and family member for their love, encouragement, care and support.

Paridhi Naithani
801532038

Abstract

Cloud computing is the emerging computer technology in the present era. It is an internet based technology and provides shared resources such as databases, storage, servers, softwares to the users as per the demand. Consumers then pay according to the usage. There are various aspects which influence the system performance and scheduling is one of them. So there is a need of an coherent scheduling algorithm that can enhance the overall performance. Scheduling algorithms mainly emphasize on completion time, cost and makespan. Furthermore there are many heuristic based algorithms for scheduling but it is observed that genetic algorithm converges faster and gives optimal results.

In this research an efficient scheduling approach is introduced which focuses on reduction of energy consumption. The overall energy which is consumed is determined by resource utilization. The proposed genetic algorithm based approach assigns the task to virtual machine according to the utilization so that overall energy consumption is minimized.

Comparison is made between FCFS, SJF and the proposed approach. Results of the evaluation work prove that our approach has least energy consumption among the three algorithms.

Keywords: Cloud computing, Virtualization, Scheduling, Genetic Algorithm.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

ACO	Ant Colony Optimization
GA	Genetic Algorithm
QoS	Quality of Service
AGA	Adaptive Genetic Algorithm
FCFS	First Come First Serve
GA	Genetic Algorithm
SLA	Service Level Agreement
FIFO	First In First Out
SJF	Shortest Job First
JLGA	Job Spanning Load Balancing Genetic Algorithm
DAG	Directed Acyclic Graph

Chapter 1

Introduction

This chapter introduces cloud computing mentioning its definition and presents the evolution of cloud computing. Further it describes the characteristics, challenges, services, types of cloud and research issues in cloud computing.

1.1 Evolution of Cloud Computing

Cloud computing is an emerging technology which made it a buzz word in the market. It has different technologies associated with it like grid computing, utility computing, distributed computing, ubiquitous computing and on-demand computing [1]. Cloud computing has evolved through many stages and grid computing being the foundation for its development. Although cloud computing has an edge over grid computing but it cannot substitute grid computing [2]. Figure 1.1 depicts that it is also based on the idea of utility computing and SaaS.

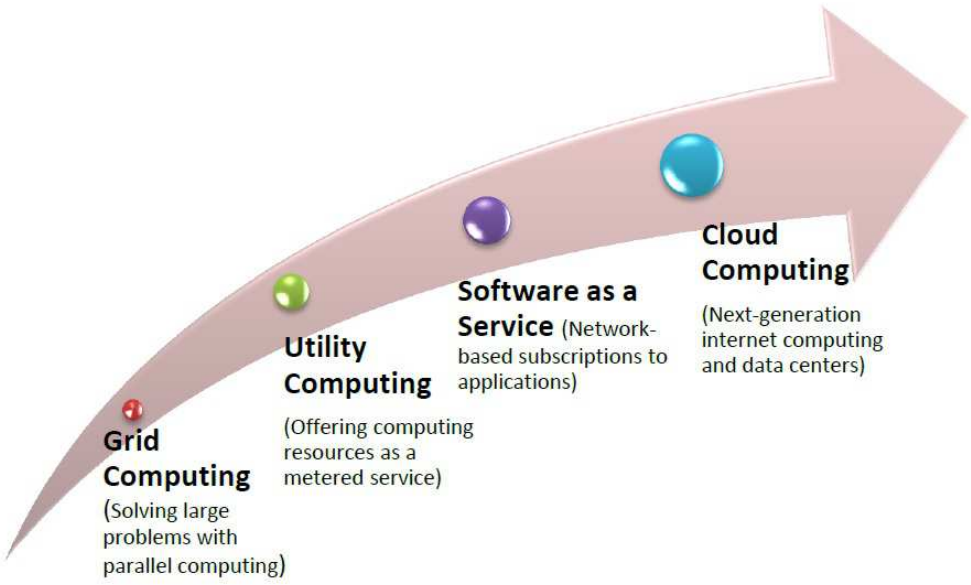

Figure 1.1: Evolution of cloud computing [1]

Figure 1.2 shows the timeline of cloud computing starting from the time of main-

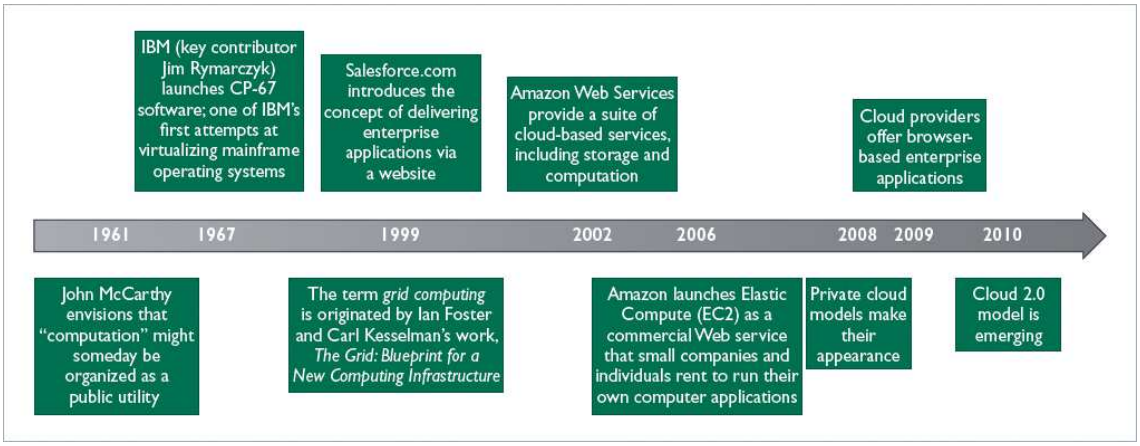

Figure 1.2: Cloud computing timeline [3]

frames. Arrival of cloud computing has added new panorama to other technologies also which are related to internet. For instance new service level agreement need to be signed between the service provider and the client. Moving to cloud has also resulted in development of new security operations. Recently cloud has been integrated with Web 2.0 and this cloud model emphasize on high performance, availability and security of infrastructure for business implementations. Cloud services are being used in different fields such as business, online gaming, academics, prediction, manufacturing. Hence cloud computing has become remarkable with time [3].

Cloud computing involves not only the services across internet but also the hardware required for storage. Cloud is basically the hardware and software of the data center [4]. Till now cloud computing does not have single canonical definition, it has been defined in different ways by different scientists. Everyone has its own perspective for cloud computing having one thing in common that user does not require its own data center. Definition of cloud computing given by NIST is as follows:

National Institute of Standards and Technology (NIST) defines Cloud computing as: "Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This Cloud model promotes availability and is composed of five essential characteristics, three service models, and four deployment models" [5]

The main aspects of cloud computing are described in the Figure 1.3.

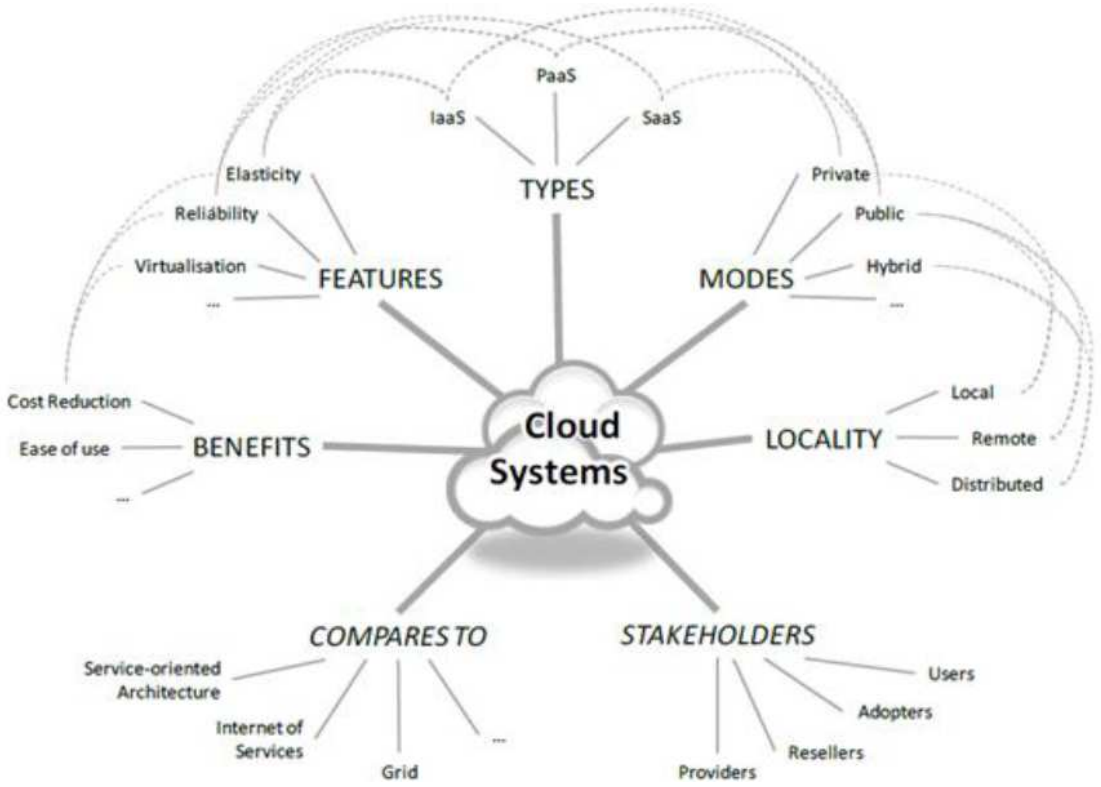

Figure 1.3: Main aspects of Cloud Computing [6]

1.2 Challenges In Cloud Computing

In the early stages, growth of cloud computing was not facile and there were several obstacles affecting it [4]. Some of these are explained below:

- **Business Continuity and Service Availability:**
  Users were resistant to move to cloud due to the fear of business continuity policies and adequate availability of services.
- **Data Lock-In:**
  Due to some inactive standardizations users could not transfer the data from one site to another. Hence this was becoming a concern for users to embrace cloud computing.
- **Data Confidentiality:**
  Security is also the prime factor leading to non adoption of cloud computing. Although it has made external security easy but internal security issues are still present. Besides this cloud user has to be protected against its provider also.
- **Data Transfer:**

The applications are becoming more data insensitive thereby complicating the placement and transfer of data. The cost of transferring data is the major bottleneck in the advancement of cloud.

- **Performance Unpredictability:**
  Although CPU and main memory sharing performance is good in cloud but the network and disk Input/Output is the concern which makes the performance unpredictable. Another factor which leads to performance unpredictability is scheduling of virtual machines.
- **Scalable Storage:**
  Application of persistent storage was not explicit in cloud computing. There was a need of storage system which not only meets the requirement of the users but also has good scalability factor.
- **Bugs in Large Scale Distributed Systems:**
  Complexity in large systems often lead to several bugs and removal of these bugs was a big challenge in cloud computing.
- **Software Licensing:**
  Software license is a restriction for the system on which software runs. In the initial phase licensing model of cloud computing was not favourable.

1.3 Characteristics of Cloud Computing

The main characteristics of cloud computing [5] are explained below:

- **On Demand Self Service:**
  Users can acquire various services like storage, network according to their requirement without interacting with the supplier.
- **Broad Network Access:**
  Users can access the services through standard policies over the network using different platforms such as phone, tablet, computers.
- **Location Independence and Resource Pooling:**
  Various resources are pooled and dynamically assigned to the users when desired. Here users need not be aware about the location of the resources and can access them irrespective of the location.
- **Rapid Elasticity:**
  Resources in cloud are rapid and elastic in nature and can scale up and down

according to the requirement. These appear to the users as an infinite pool and can be acquired on demand.

- **Measured Service and Reduced Cost:**
  Resource usage is transparent to both the user and the provider and users get the resources on pay as per usage base. The initial setup cost can be avoided as the purchase of infrastructure is not required. So the overall cost is reduced in cloud computing.

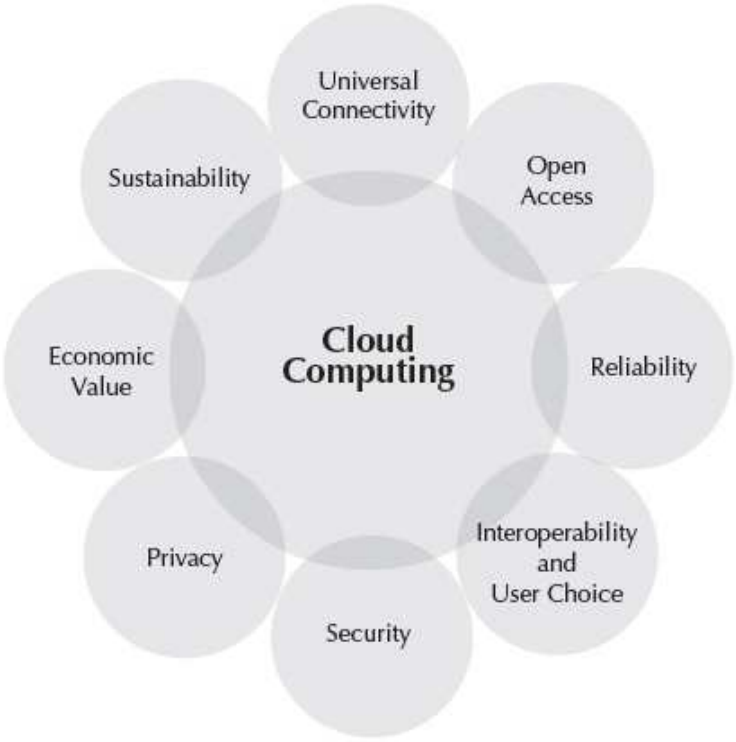

Figure 1.4: Features of Cloud Computing [1]

1.4 Cloud Computing Services

The services are broadly classified into three categories [7] as exhibited in the Figure 1.5

(i) **Software as a Service (SaaS)**

- SaaS model distributes software and service providers host the applications and delivers it to the users over the internet.
- It is implemented to provide services at low cost without the need of initial setup which can be accessed from any location.

- The softwares are used on pay as per use policy as customers use the license of the provider rather that purchasing their own.
- It supports more than one customer at the same time and hosts one application for multiple customers.
- Google Apps, Salesforce.com are the major SaaS providers.

(ii) **Platform as a Service (PaaS)**

- PaaS provides a development environment to the consumers where they can create and deploy their application.
- Here platform include libraries, tools, programming languages and services which can be used by the users for the development of applications.
- The service offered to the consumers include the complete software development life cycle.
- Management of the cloud resources like network, storage, servers and operating system is done by the provider.
- Leading PaaS providers are Google App Engine, Windows Azure.

(iii) **Infrastructure as a Service (IaaS)**

- It is a cloud layer where consumers are provided services like storage, network and software resources like operating system where the user can deploy and run its software applications.
- Cloud resources are assigned to contracted customers on pay per usage base. This reduces the overall cost as the need of initial infrastructure setup including hardware like server and network is waived off.
- Moreover it makes the system more flexible as the resources can be scaled up and down rapidly.
- Users do not even need to take care of the maintenance, provisioning of the infrastructure and other complexities.
- The main IaaS providers are Windows Azure, Google Compute Engine and IBM SmartCloud Enterprise.

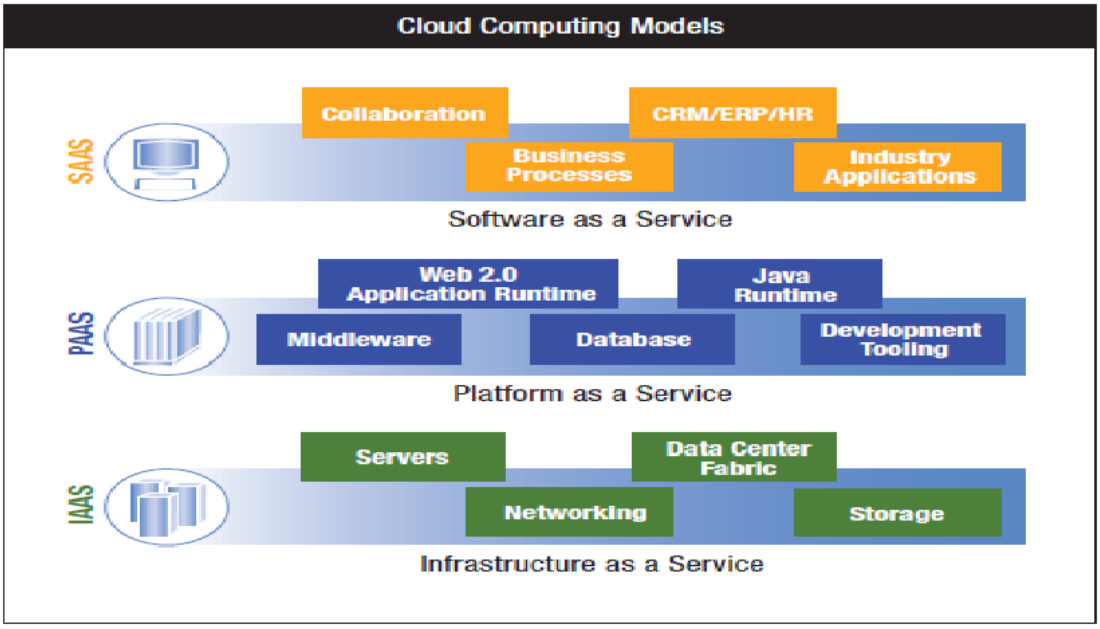

Figure 1.5: Cloud Computing Services [7]

1.5 Cloud Computing Deployment Models

Deployment models depict the cloud environment and vary according to size and access. Different deployment models [7] are explained below:

(i) **Public Cloud**

- It is a cloud hosting model that provides services through internet to the public.
- It is an economical model based on pay as per go and user need not be aware about the location of the resources.
- The main difference between public and private cloud is the security issue.
- It is suitable for the needs which require load management and for SaaS based applications.
- The major public clouds are Amazon Elastic Compute Cloud (EC2), IBM's Blue Cloud, Sun Cloud, Google App Engine.

(ii) **Private Cloud**

- Internal cloud is another name for private cloud. It is implemented using a firewall within a particular organization for its internal users.

- It is more secure than public cloud as the access is granted only to internal members of the organization.
- It is more suited for the organisations having unpredictable needs, critical missions and uptime requirements.

(iii) **Hybrid Cloud**

- It is an integration of two or more cloud models one of them being private cloud and rest are the public cloud. It combines the advantages of both public and private clouds.
- It is centrally managed and more secure. The organization keeps the sensitive data and outsource the non critical information to the public cloud.
- It is adopted by businesses having security and demand as their main concerns.
- Scalability, flexibility and security are some of the prime features of hybrid cloud. It can also be used for processing of big data.

(iv) **Community Cloud**

- Community cloud is a type of cloud model having multi-tenant setup which is shared by different organisations belonging to same community having similar concerns.
- The management and hosting can be done either externally or internally.
- Overall cost is minimized as it is split among different organisations of the community.
- It is implemented by the businesses having similar goals working on common ventures.

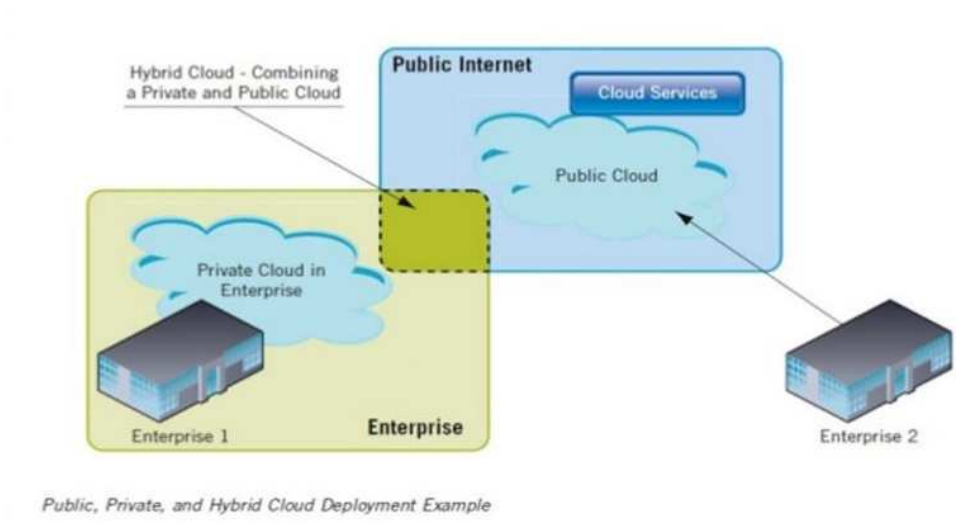

Figure 1.6: Cloud Computing Deployment Models [8]

1.6 Virtualization

1.6.1 Overview of Virtualization

The virtualization concept has become popular in cloud environment due to its reduced cost and appropriate management in resource sharing. Virtualization allows to create virtual form of anything like server, desktop, storage or operating system. Basically in virtualization more than one customer can share single physical resource. For instance multiple operating systems can run simultaneously on a single physical machine. It logically divides the resources and assigns logical name to the physical resource which is pointed when required. Virtualization partitions the physical resources into virtual machines representing virtual hardware which executes on hypervisor as shown in the Figure 1.7

1.6.2 Characteristics of Virtualization

The main characteristics of virtualization [10] which makes it ideal for cloud computing are :

- **Partitioning:** Single physical system is partitioned to assist multiple operating systems and applications.

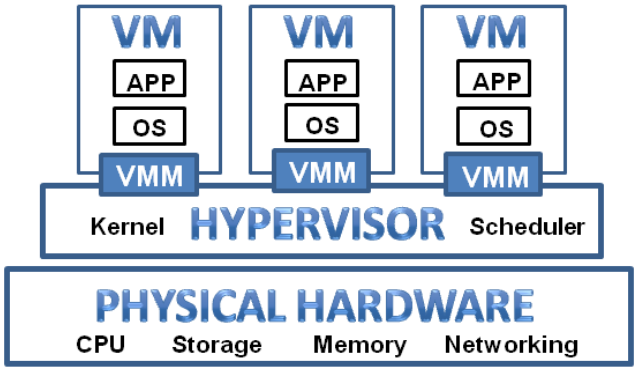

Figure 1.7: Representation of virtualized environment [9]

- **Isolation:** Virtual machines are isolated from each other and also from its host machine due to which machines do not interfere and failure of one machine do not affect the other. Sharing of data between virtual machines is also not allowed.
- **Encapsulation:** Each virtual machine is encapsulated as a single entity and can be identified on the basis of its service. Thus encapsulation also ensures that applications do not interfere with each other.

1.6.3 Need of Virtualization

Due to many benefits of virtualization, it has become a necessitate factor for the cloud environment [11]. The reasons for its need are explained below :

- Requirement of less hardware, less cost, better management makes virtualization a beneficial concept for cloud computing.
- Multiple under utilized machines can be consolidated to a single machine, hence balancing the workload among them.
- Several applications cannot run on a single execution environment and this is not possible without virtualization. So legacy applications are managed better in virtualized environment.
- Security is the major concern which is handled well by virtualization through the implementation of secure platform. Address obfuscation can also be used to run untrusted applications.
- Virtualization allows to create execution environment with limited resources while maintaining the QoS.

- Virtualization makes simulation of network of computers, migration and debugging easy.

1.6.4 Types of Virtualization

- **Hardware Virtualization**
  It creates a virtual machine that serves like an actual system having its own operating system. For example a system having Linux operating system can host a virtual machine having Xen operating system. Host machine creates guest machine over it which is known as virtual machine. Hypervisor also known as Virtual Machine Manager is the software that creates virtual machine on the physical machine. Multiple small physical machines are combined to a single large physical machine to make the best use of it [12].

  It is further categorized as follows:

  - *Full Virtualization*: It allows simulation of whole hardware to run unmodified guest operating system.
  - *Para Virtualization*: It needs modification of the guest programs to run as a separate system in the isolated domain while the hardware system is unmodified.
  - *Partial Virtualization*: In this hardware system need to be modified.

- **Desktop Virtualization**
  It is the most common type of virtualization and beneficial for IT people. In this desktop of the user resides on the remote server and can be accessed from any system or location. As per the convenience, user can also work from home. It provides data security by keeping data safe on central server, hence minimizing the risk of data loss [13].

- **Network Virtualization**
  It manages the computer network as a single unit from a single administrator. In this, more than one small networks can be constructed on a single physical network. It aims towards the optimization of data transfer rate, security and scalability. It is reliable as file transfer is restricted and one network does not affect the other. Two types of network virtualization are:

  - *Internal*: It allows single system to work as a network.
  - *External*: Several sub-networks are combined to one or single network is partitioned into many.

- **Storage Virtualization**
  In storage virtualization more than one storage resources are merged to form single storage system. It results in several advantages like reduction of downtime, effective management of storage resources, optimal performance and homogeneity across storage resources.

  It is further classified as:

  - *Block*: Multiple storage units are combined to a single unit.
  - *File*: It provides access to remote files hosted over multiple systems.

- **Memory Virtualization**
  It aggregates physical memory of multiple servers to form one virtual memory thereby enlarging the working memory. It provides benefit of shared, distributed and networked memory.

  Subtypes of memory virtualization are:

  - *Application-level*: It provides direct access to the memory.
  - *Operating system level*: In this, memory can be accessed through an operating system.

- **Software Virtualization**
  Software virtualization hosts multiple virtual environments on a single physical machine. It enables complete computer system to run guest operating system.

  Different categories of software virtualization are:

  - *Operating system*: It hosts more than one operating system on a native operating system.
  - *Application virtualization*: It hosts multiple applications in the virtualized system
  - *Service virtualization*: It hosts processes and provides services related to it.

1.7 Research Issues In Cloud Computing

Although arrival of computing has been a boon for IT industry but still there are some research issues that are not completely addressed [14]. Some of the research issues are discussed below:

(i) **Automated Resource Provisioning**

In cloud computing allocation and deallocation of resources according to the need are the essential factors. The main aim of the provider is to assign and release the resources to fulfill the SLA while keeping the cost low. So, in order to satisfy the demands effective resource provisioning is required. There are several approaches for automated resource provisioning such as:

- Application performance model can be created which predicts the number of application samples required to fulfill the demand and QoS.
- Use performance model to forsee demands and resource needs.
- Queuing model can be used to create application model and automatically assign resources on the basis of resource prediction.
- The proactive approach uses resource prediction to assign resources prior to the need.
- The reactive approach immediately fulfills the demand prior to periodic prediction.

(ii) **Virtual Machine Migration**

Virtualization is a fundamental concept in cloud computing that enables virtual machine migration for load balancing. Moreover virtual machine migration leads to vigourous and quick receptive provisioning. In recent times live migration of virtual machines has been executed by Xen and Vmware, minimising the downtime to 10 milliseconds. In process level migration the complete operating system and its applications should be migrated as a single component. Virtual machine migration has resulted in the evasion of hotspot and this becomes its main advantage.

(iii) **Server Consolidation**

Server consolidation is a productive mechanism in cloud computing as it minimizes energy expenditure and maximizes the resource utilization. Live migration consolidates underutilized servers into one server and sets the leftover servers to save energy state. Server consolidation should be done in a such a way that the application performance is not affected. Resource usage also known as footprint, changes with time and the sharing of resources among different machines may lead to congestion. So the system should be highly responsive to such congestions.

(iv) **Energy Management**

Efficient energy management is the major concern in cloud computing. It is observed that 53% of the total expenses are used for power and cooling con-

cerns. Energy consumption is increasing at a rate of 18% every year, hence there is need of effective infrastructure. Energy expenditure in the data center must be cut while maintaining the government laws and environment norms. Energy-saving data centers can be constructed using different approaches. Common method is to use power effective hardware that lowers the CPU speed and turns off the unused machines. Further energy aware assigning of tasks and server consolidation can also be used. All the approaches should be applied without affecting the overall performance of the application.

(v) **Traffic Management and Analysis**
Traffic analysis is required in different fields nowadays. Many web applications analyse traffic to improve user experience. Network operators also keep themselves aware about the traffic flow so as to manage and plan the decisions. But traffic management and analysis in data centers is not staightforward. Existing methods have certain limitations as they can manage only hundreds of servers whereas data centers may have thousands of hosts. Also these methods work for only few assumed patterns.

(vi) **Data Security**
Apart from above mentioned issues in cloud computing, data security is also a vital research topic. Service providers count on infrastructure providers to attain data security as they cannot access the data center directly. Service providers define remote security settings for private cloud also. The infrastructure supplier accomplish confidentiality using cryptographic protocols for secure data transfer and access and auditability is established using remote attestation techniques to identify if security settings are altered. Remote attestation is done using trusted platform module (TPM) to produce system summary to validate the security of the system. However remote attestation is not adequate in case of dynamic migration of virtual machines. In this case each layer, hardware layer, virtual platform need to build trust mechanisms. Currently research is done to establish effective protocols for security.

(vii) **Novel Cloud Architectures**
Implementation of commercial cloud in large data centers achieve scalability and management but has certain constraints with it like high initial cost for creation of data center and excessive energy costs. Recent work propose that small data centers are beneficial than larger ones as their power consumption is less, economical to construct and better geographically scattered. Other way can be to use voluntary or a mix of voluntary and dedicated resources

as they are cost effective and favourable for scientific computing. Besides having these advantages these architecture have management issues.

1.8 Structure of Thesis

The rest of the thesis is organised as follows:

Chapter 2- This chapter discusses the review of literature survey done related to the work including virtualization, scheduling, existing task scheduling algorithms and their comparison.
Chapter 3- This chapter gives the brief overview of the problem in addition to analysis of the gaps found during the literature survey.
Chapter 4- This chapter describes the proposed solution in detail along with its algorithm.
Chapter 5- This chapter emphasize on the implementation of the algorithm in cloudsim environment. It also presents the evaluation work and results obtained.
Chapter 6- This chapter concludes the thesis work with its contribution and future scope.

Chapter 2

Literature Review

This chapter discusses the review of literature survey done related to the work including scheduling, existing task scheduling algorithms and their comparison.

2.1 Scheduling In Cloud Computing

Assigning of tasks to the virtual machines as per the requirements is called scheduling[15]. Scheduling in cloud computing has some constraints associated with it which can be related to time or resources [16]. In real time many applications are to be completed within the deadline using minimum resources. Hence efficient task scheduling [17] and resource scheduling policies are required. The aim of scheduling algorithm is to allocate tasks such that load is balanced and resource utilization is maximal. These algorithms also try to optimize certain objective function. Figure 2.1 shows the mapping between virtual machines and the physical systems.

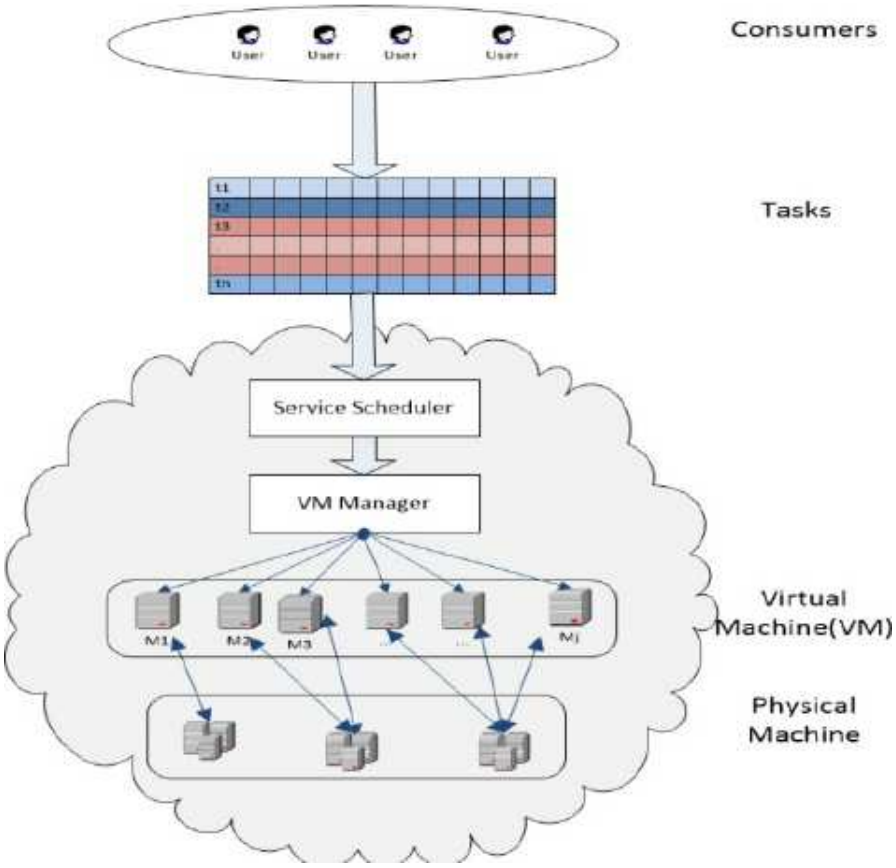

Figure 2.1: Scheduling Architecture of Cloud Computing [18]

2.1.1 Types of Scheduling

- **Static Scheduling**
  In static scheduling task related information like execution time and mapping is known before execution [19]. Runtime overhead in case of static scheduling is less.

- **Dynamic Scheduling**
  In this task related information is not known and assignment of tasks is done at the time of execution of task [20].

- **Heuristic Scheduling**
  It is a type of scheduling that schedules the task based on heuristic function and finds near optimal solution at a faster rate.[21]

- **Workflow Scheduling**
  In workflow scheduling applications are structured as a workflow where nodes represent tasks and edges depict the dependencies between the tasks. It finds suitable resources and assigns tasks on these [22].

2.2 Existing Task Scheduling Algorithms

In this section, brief description of various task scheduling algorithms is given. Task scheduling algorithms are classified into three broad categories simple, heuristic and metaheuristic algorithms as described below:

(i) **Simple Scheduling Algorithms**

- **First Come First Serve (FCFS):**
  FCFS considers arrival time of tasks which means task which arrives first will be executed first. It is non-preemptive in nature. Although it is not a complex algorithm but in some cases it leads to inefficiency. For instance if a longer task arrives first then the shorter task which arrives later will have to wait for its execution, increasing the waiting time of shorter tasks [23].
- **Round Robin Algorithm:**
  It dispatches the tasks in FIFO manner but it assigns the resources to each task for fixed time quantum and if the task is not completed before the time slot allocated then resources are preempted until its next turn for execution and allocated to the next waiting task. It is easy to implement, distributes resources evenly and independent of starvation but it is critical to decide the time quantum [24].

(ii) **Heuristic Based Task Scheduling Algorithms**
Heuristic algorithms [25][26] yield optimal solutions using knowledge base. Some of them are:

- **Minimum Execution Time:**
  This strategy schedules the tasks based on their predicted execution time. It selects the best machine on which the tasks take least time but does not consider availability of resources which sometimes lead to load imbalance [27].
- **Minimum Completion Time:**
  It considers predicted completion time of the tasks as the scheduling parameter. It randomly allocates task to the machine having minimum completion time. In this algorithm some tasks may get scheduled to the machine that does not have minimum execution time [27].
- **Max-Min Algorithm:**

It firstly creates expected execution time and expected completion time matrices. It schedules the task having maximum completion time on the best machine available with least execution time. It also avoids starvation by executing smaller tasks on other machines in parallel with longer tasks. It increases the makespan and utilization of resources [28][29].

- **Min-Min Algorithm:**
  This algorithm begins with a set of tasks and allocates the smallest task to the machine with minimum completion time for all tasks. Then it removes this task from the set and same steps are iterated until all the tasks are scheduled [30][31].

- **Earliest Deadline First:**
  It is also known as Least time to go. It is a dynamic scheduling strategy which places the tasks in a priority queue. Task with close deadline is given higher priority and is assigned to the machine first. It is used in real time operating systems [32].

(iii) **Meta-heuristic Scheduling Algorithms**
These are also known as approximation algorithms. Meta-heuristic algorithms are better than traditional scheduling algorithms [33][34] as they find solution iteratively in a reasonable time.

- **Simulated Annealing:**
  It is derived from physical annealing process of obtaining low crystalline state os solid. Finally an optimal state of thermal equilibrium will be reached which corresponds to the optimal mapping of tasks in scheduling. It is an iterative process and is initialized with a single solution. New state is evaluated and then compared with the previous state. Old state is replaced with the new state if it is better than the old [35].

- **Tabu Search:**
  Tabu search is a technique which keeps the record of the visited regions in the search space so as to avoid repetition. It is implemented with the use of memory structures. It iteratively moves from one solution to the other neighbouring solutions until termination condition has been satisfied with an aim to minimize some objective function [36].

- **Genetic Algorithm:**
  It is a evolutionary technique based on the concept of "survival of the

fittest". It is divided into four main steps: population generation, chromosome evaluation, crossover and mutation, evaluation of modified chromosome. The above steps make one iteration of the algorithm and number of iterations are fixed according to the problem. These steps are iteratively performed until a stopping criteria is met [37].

- **Particle Swarm Optimization (PSO):**
  PSO [21] is a population based stochastic algorithm inspired by flocking of birds and schooling patterns of fish. Similar to GA it starts with random population generation but has no genetic operators like crossover and mutation. In PSO solutions are known as particles which fly through multidimensional space following current optimum particles. It is easy to implement and has less parameters to adjust. It merges both local and global search method.

- **Ant Colony Optimization (ACO):**
  It is biological inspired probabilistic algorithm which finds good solutions through graphs. It is similar to behaviour of ants looking for food in their colonies through pheromone. It can be applied to any combinatorial problem like scheduling problem, vehicle routing problem, assignment problem, image processing and others. Although it has some advantages like use of positive feedback mechanism, extensible nature but at the same time it converges to local optimal solution [38].

- **Fuzzy Algorithms:**
  Fuzzy neural networks are used for optimal scheduling these days. Neural network is a data driven tool which represents non linear relationships. [39]

Table 2.1: Comparison of Existing Scheduling Algorithms

Algorithm	**Parameter**	**Advantages**	**Disadvantages**
First Come First Serve (FCFS)	Arrival Time	Easy to implement	Non-preemptive and does not consider any criteria for scheduling
Round Robin	Arrival time, Time quantum	Fair Load balancing	Size of time quantum

Minimum Execution Time	Expected execution time	Selects fastest machine for scheduling	Load is not balanced
Minimum Completion Time	Expected completion time	Proper load balancing	Resource selection is not optimal
Max-Min,Min-Min	Expected completion time	Works for makespan and resource utilization	Load balancing and Qos is not considered
Earliest Deadline First	Deadline	Optimal for preemptive uniprocessors	It does not work well when the system is overloaded
Simulated Annealing	Makespan	Easy to code and provides good solutions	In some cases it takes huge time to produce results
Tabu Search	Completion time, Makespan	Good for large and difficult problems	Number of iterations can be large
Genetic Algorithm (GA)	Makespan, Energy, Cost, Efficiency	It can be used for complex objective function	No guarantee of global maxima
Particle Swarm Optimization (PSO)	Makespan, Cost, Performance	It is simple and fast than other algorithms	It suffers from partial optimization

2.3 Related Work

This section briefly investigates recent work done related to scheduling in cloud considering scheduling parameters, technique and tools used.

- **Scheduling based on Berger Model:**
  Baomin Xu et al. [40] have implemented a job scheduling algorithm based on Berger Model considering the commercialization and virtualization characteristics of cloud computing. In this algorithm dual fair constraint for resource selection and resource allocation is created. The proposed algorithm has been implemented in ClouSim.
- **Dynamic Trusted Scheduling:**

Wei Wang et al. [41] have proposed cognitive trust model based on Bayesian cognitive model. Further authors have integrated DLS algorithm to propose Cloud-DLS which is a dynamic trusted scheduling algorithm. Simulations are performed on CloudSim environment and results prove that this algorithm execute tasks securely considering the execution time and reliability.

- **Cost Effective Task Scheduling:**
  Su et al. [42] have presented two heuristic approaches to implement cost effective task scheduling algorithm for large programs. One is based on Pareto Dominance that dynamically map tasks to the best virtual machine in terms of cost. Second strategy minimizes the monetary cost of non critical tasks. Experiments are performed on randomly generated large DAGs. Results show that it has substantially reduced the cost and produces fair makespan.

- **Deadline-constrained Workflow Scheduling Algorithms:**
  Saeid Abrishami et al. [43] have proposed two workflow scheduling algorithms for IaaS cloud based on Partial Critical Paths (PCP) algorithm. Authors work have reduced the execution cost of the workflows while satisfying the deadlines. Both algorithms perform well for scheduling of large workflows.

- **Scheduling for Optimal Service Deployment:**
  Jose et al. [44] developed broker framework for multi cloud environment. Different optimization criteria like cost and performance optimization have been explored. Evaluations are done on HPC cluster and web cluster under different conditions.

- **Resource Aware Scheduling Algorithm:**
  Saeed and Reza [45] designed an algorithm named Resource Aware Scheduling algorithm based on Min-Min and Min-Max algorithms. It has achieved lower makespan in distributed systems.

- **Reliable Scheduling Distributed in Cloud Computing (RSDC):**
  Arash et al. [46] created an algorithm for reliable scheduling in cloud environment. Besides other parameters, they have considered request time and acknowledge time of tasks. It takes less processing time in cloud as compared to other algorithms.

- **Priority based Service Scheduling Policy:**
  Dakshayini and Guruprasad [47] suggested a scheduling approach which satisfies the QoS. It is based on priority and admission control and optimizes the time spent by the task in the queue. It has achieved high throughput

and very high service completion rate.

- **Energy-efficient Scheduling of HPC applications:**
  Saurabh Kumar Garg et al. [48] propose energy based scheduling strategies which explore heterogeneity among different data centers. Several energy efficient parameters are considered like energy cost, workload and carbon emission rate. These policies have gained upto 30% energy saving and also reduced carbon emissions.

- **Power-aware Provisioning of Cloud Resources:**
  Kyong Hoon Kim et al. [49] investigated realtime cloud framework. Three power aware provisioning policies have been discussed namely Lowest DVFS, Adaptive DVFS and Advanced DVFS VM provisioning. These DVFS schemes have resulted in less power consumption and increased profit. Simulations have been performed on CloudSim kit.

- **Scheduling of Scientific Workflows:**
  Golnar Gharooni-fard et al. [50] established chaos genetic algorithm which uses chaotic series. To schedule tasks to virtual machines it considers both users budget and deadline. It can be applied to balanced and unbalanced workflows. Evaluation results show its better performance as compared to simple genetic algorithm due the chaotic feature which scatters the solution in whole search space.

- **Load balancing Task Scheduling:**
  Tingting Wang et al. [51] introduced new scheduling approach named JLGA based on AGA. It has adapted greedy algorithm for population generation and uses multi fitness function. It also satisfies load balancing among different nodes. It is compared with AGA and experiment results show that average makespan is less for JLGA and efficiently balances system load.

- **GA-based Task Scheduler:**
  Yujia Ge et al [52] designed new algorithm to schedule the tasks based on GA. It evaluates group of tasks present in the queue to schedule the tasks. In comparison to FIFO and delay algorithms it has shorter makespan and balances load among the nodes.

- **Energy-efficient Management of Data center Resources:**
  Buyya et al. [53] have developed algorithms for allocation and provisioning of cloud resources that improves energy efficiency while maintaining the QoS requirements without violating SLA parameter. The authors have presented policies for scheduling of virtual machines to acceptable resources along with

dynamic consolidation of virtual machine resources. These approaches have been implemented and simulated on the CloudSim environment.

- **Optimizing Energy Consumption in Clouds:**
  Sahar Hosseinzadeh, Mirsaeid Hosseini Shirvani [54] have investigated the problem of energy in task scheduling. A genetic algorithm was proposed to allocate tasks to virtual machines on the basis of availability and energy consumption of the resources. Virtual machines having all the necessary resources which also consume least energy are used. Result evaluation have shown that this algorithm reduces the costs and the energy consumption.

Table 2.2: Comparative Analysis

Algorithm/ Technique	**Parameter**	**Findings**	**Simulation Tool**
Job Scheduling algorithm based on Berger Model[40]	Completion time, Bandwidth	(i) Effective execution of tasks. (ii) It manifests better fairness.	CloudSim
Dynamic trusted Scheduling [41]	Exection Time	(i) Execute tasks securely. (ii) Scalable for diverse application domains.	CloudSim
Cost-Effective Task Scheduling for large programs [42]	Cost, Makespan	(i) Substantial reduction in monetary costs.	CloudSim
Deadline-constrained Workflow Scheduling Algorithms [43]	Cost, Deadline	(i) Exectue tasks before the deadline. (ii) Suiltable option for large workflows.	Cloud environment

Scheduling strategies for optimal service deployment [44]	Cost, Performance	(i) Overall performance is improved.	Cloud environment
Resource Aware Scheduling Algorithm (RASA) [45]	Makespan	(i) Reduced makespan. (ii) Better than existing algorithms for large scale distributed systems.	GridSim
Reliable Scheduling Distributed in Cloud Computing (RSDC) [46]	Processing time	(i) It minimizes the processing time in cloud.	Cloud environment
Priority based Service Scheduling	Service request time	(i) High service completion time. (ii) Satisfies Quality of Service.	Cloud environment
Cost based algorithm for task scheduling [55]	Cost, Performance	(i) Improved computation/communication ratio. (ii) Satisfies Quality of Service.	Cloud environment
Energy-efficient Scheduling of HPC applications [48]	Energy cost, CPU power efficiency	(i) Energy savings upto 30%. (ii) Less carbon emissions.	Cloud environment

Power-aware Provisioning of Cloud Resources [49]	Power consumtion	(i) Less power consumption. (ii) More profit.	CloudSim
Scheduling of scientific workflows [50]	Users budget, deadline	(i) Greater performance than simple Genetic Agorithm.	Grid environment
Load Balancing Task Scheduling [51]	Makespan, Load balancing	(i) Efficient inter node load balancing. (ii) Less makespan than AGA.	Matlab
GA-Based Task Scheduler [52]	Makespan	(i) Shorter makespan.	Java with GA package, JGAP
Energy-efficient management resources [53]	Quality of Service (QoS), SLA	(i) Improves energy eficieny.	CloudSim

Chapter 3

Problem Analysis

This chapter emphasize on exploration of gaps found during literature survey of different scheduling techniques. It also discusses the problem statement and its description.

3.1 Gap Analysis

Based on the literature survey different energy based scheduling policies have been described in Table 3.1 . Although these techniques are energy efficient but still have some drawbacks and do not provide optimized scheduling approach to minimize energy consumption.

Table 3.1: Energy Efficient Scheduling Techniques

Technique	**Findings**
Adaptive energy-efficient scheduling [56]	Reduces voltage levels to conserve energy
Phase-aware scheduling for multicore processors [57]	Energy minimization over random scheduling of programs
Job scheduling model based on GA [58]	Less energy consumption and higher profit
A metaheuristic for energy aware scheduling [59]	Energy saving based on DVS technique
Online self reconfiguration for energy efficiency [60]	Improved resource utilization and conserve energy by switching off more unnecessary machines
Power aware virtual machine allocation [60]	Longer computational time but lower energy consumption
Energy aware workload placement [61]	Better server utilization leads to superior energy gains

Phase change memory optimization [62]	Improves the system performance and reduces the maximum memory usage
Pareto solution based GA [63]	Effective convergence, stability and solution diversity

3.2 Problem Statement

Energy efficiency is a crucial factor in cloud computing. To address this issue a simple and efficient scheduling approach which lowers the overall energy consumption need to be implemented. Problem here is how tasks should be scheduled so as to make utmost use of resources thereby reducing the energy consumption.

3.3 Problem Description

Considering different algorithms for scheduling it is observed that GA is an efficient algorithm for scheduling of tasks which produces decent results. It is seen that energy consumption is one of the prime issue which impacts the overall system and the efficiency of scheduling algorithm. So it is required that user selects the virtual machine that meets the necessary conditions. In this thesis, main aim is to schedule the task such that overall energy consumption is reduced.

The problem can be formally defined as:

- Let T = { T_1, T_2, T_3, ...,T_x } be a task set consisting of x tasks that need to be scheduled.
- Let V = { V_1, V_2, V_3, ...,V_y } be a set of virtual machines on which task will be mapped.

Now the problem is to map these x number of tasks to y machines such that overall energy is reduced. So to find a solution to this problem GA is designed described in next chapter.

3.3.1 Objective

The main intent of this research is to schedule the tasks to virtual machines in such way that:

- Resource utilization is maximum.
- Overall energy consumption is reduced.

Chapter 4

Proposed Solution

This chapter describes the proposed solution to the problem stated in the previous chapter. It also illustrates the design and detailed description of the task scheduling technique.

4.1 Design of Proposed Scheduling Technique

The proposed scheduling technique is GA based that assigns tasks to virtual machines in an efficient manner. GA has proved to be successful evolutionary algorithm for scheduling which provides suboptimal solution in acceptable time [64]. Task scheduling is carried out based on the resource utilization so that the overall energy consumption is minimized.

4.2 Genetic Algorithm Based Task Scheduling

This section describes the GA based task scheduling. Further it also explains the steps of the algorithm in detail starting from encoding scheme, fitness function and genetic operators used.

Algorithm 4.1 described in this section begins with initializing the population to generate feasible random solutions. Fitness value of each solution is calculated and stored. In order to produce better solution parents are selected to apply genetic operators like crossover and mutation. Best solutions are saved and these steps are iterated until stopping condition is met.

Algorithm 4.1 Workflow of GA

1: Generate initial population.
2: Calculate fitness value of each chromosome.
3: **while** Stopping condition is not fulfilled **do**
4: Select parents using 4 way tournament selection strategy.
5: Perform one point crossover between the selected parents if crossover rate is met.
6: Perform mutation using random resetting.
7: Evaluate the new child.
8: Generate new population
9: **end while**

4.2.1 Chromosome Encoding scheme

In GA chromosome represents a solution of a problem. Basically chromosome is composed of number of genes. Here each chromosome is exhibited by a one dimensional array having length equal to total tasks. It is made up of x genes, where x is total number of tasks and each gene value corresponds to a virtual machine on which task is assigned. Numerically it is positive integer between 1 and y where y is the total number of virtual machines available. Figure 4.1 shows an example of task placement and its corresponding chromosome.

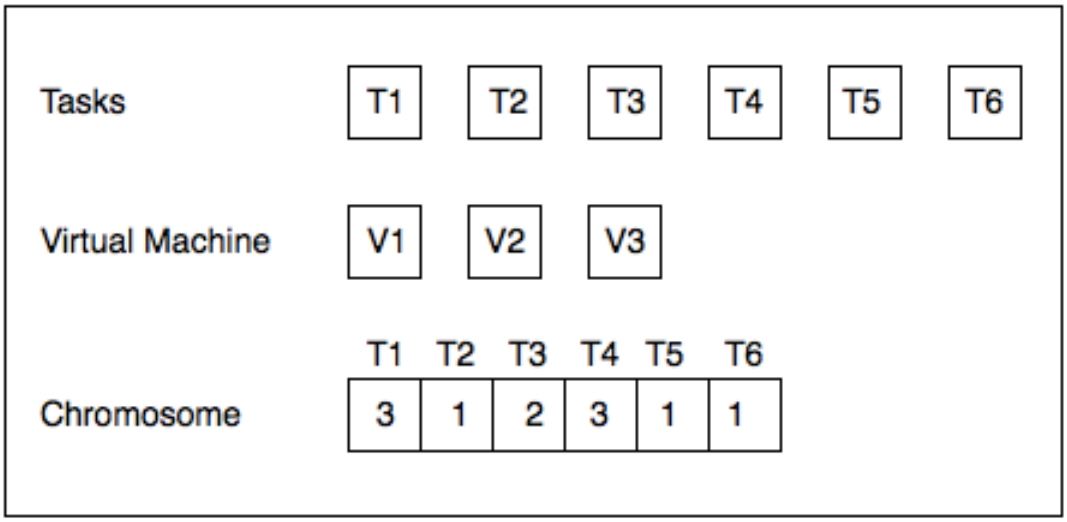

Figure 4.1: Example of chromosome structure

4.2.2 Fitness Function

Fitness function is of great significance for GA. It is a benchmark to identify if the solution is more fit or better than other solutions corresponding to the problem in consideration. The computation of fitness function should be fast enough as it is calculated repeatedly in the algorithm. Fitness function in this research is related to resource utilization [65].

Let Utilization of a resource be defined as

$$U_x = \sum_{y=0}^{p-1} (u_{x,y}) \tag{4.1}$$

where p is total tasks and $u_{x,y}$ is resource usage of task t_y

The fitness function for the proposed algorithm can be defined as follows :

$$f = \sum U \tag{4.2}$$

Algorithm 4.2 Algorithm for Fitness Function

```
1: Initialize fitness ← 0
2: for a=0 to clength do
3:     f=cloudletList.get(a).getUtilizationOfCpu(cloudletList.get(a).getActualCPUTime());
4:     fitness += f
5: end for
6: return fitness
```

4.2.3 Initial Population

Initial population is considered as a set of chromosomes. Size of initial population depends on the problem requirement. In this approach, we have used random method to generate initial population. In random method chromosomes are generated randomly from the possible set of solutions.

Algorithm 4.3 Algorithm for Initial population generation

Input: Population size, length of chromosome

Output: list of chromosomes

```
1: for i=0 to popsize do
2:     for j=0 to chlength do
3:         chromosome ← rand.nextInt()%totalVms
4:     end for
5:     population.add(chromosome)
6: end for
```

4.2.4 Parent Selection

Parent selection is also one of the key factor on which the performance of the algorithm depends. In this few individuals are selected to produce new offsprings. Fitness based selection selects those parents which have better fitness value than others. Here we have experimented with 4 way tournament selection strategy as shown in Algorithm 4.4 in which 4 individuals are selected at random and then parallel tournaments are run among them. From these 2 individuals who win the

tournament are selected for further steps. Tournament selection is an efficient method for selection and also has several advantages over the other methods.

Algorithm 4.4 Algorithm for Parent Selection

Input: 4 chromosomes

Output: 2 best chromosomes

1: $child1 \leftarrow$ (rand.nextInt()%popsize))
2: $child2 \leftarrow$ (rand.nextInt()%popsize))
3: $child3 \leftarrow$ (rand.nextInt()%popsize))
4: $child4 \leftarrow$ (rand.nextInt()%popsize))
5: $f1 \leftarrow$ getfitness(child1)
6: $f1 \leftarrow$ getfitness(child2)
7: $f1 \leftarrow$ getfitness(child3)
8: $f1 \leftarrow$ getfitness(child4)
9: **if** f1 $>$ f2 **then**
10: child1=child3;
11: **end if**
12: **if** f2 $>$ f4 **then**
13: child2=child4;
14: **end if**

4.2.5 Crossover

Crossover is used to inculcate some variation in the chromosomes. Basically it is a selection technique which selects more than one parent chromosome and produces one or more new offspring using the genes of the parents. Probability of crossover is usually kept high. Different crossover techniques exist like one point crossover, two point crossover and multi point crossover. In the proposed algorithm one point crossover is used which is explained in the Algorithm 4.5

Algorithm 4.5 Algorithm for Single Point Crossover

Input: 2 best selected parents

Output: 1 child

```
1: piv ← rand.nextInt()%(chlength − 1)
2: for i=0 chlength do
3:    if i < piv then
4:       child5.add(child1.get(i))
5:    else
6:       child5.add(child2.get(i))
7:    end if
8: end for
```

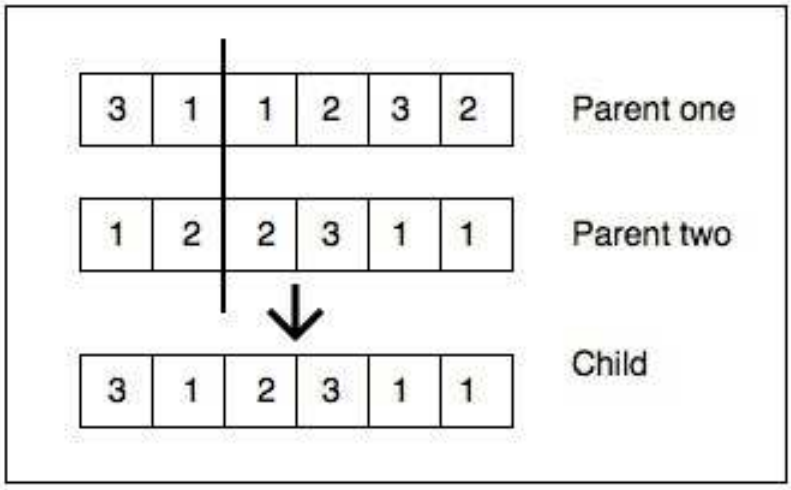

Figure 4.2: Crossover

4.2.6 Mutation

Mutation is a genetic operator that sustains population diversity, hence widening the search space. It is usually applied with low probability to prevent it from getting stuck in local minimum. Mutation leads to the convergence of GA and finds optimal solution rapidly. From various available mutation techniques experiments are done using random resetting described in Figure 4.3, which is an extension of bit flip mutation. Here random selection of a chromosome and its gene is done. This selected gene of the chromosome is replaced with a random value from a set

admissible values.

Algorithm 4.6 Algorithm for Mutation

Input: Randomly selected parent

Output: Mutated child

1: *mutationpoint* ← rand.nextInt()%popsize
2: *mutateVM* ← rand.nextInt()%totalVms
3: mr=rand.nextInt(1)
4: **if** mr < 1 **then**
5: child1.set(mutatePoint, mutateVM)
6: **end if**

Example for mutation using random setting is shown in the figure 4.3.

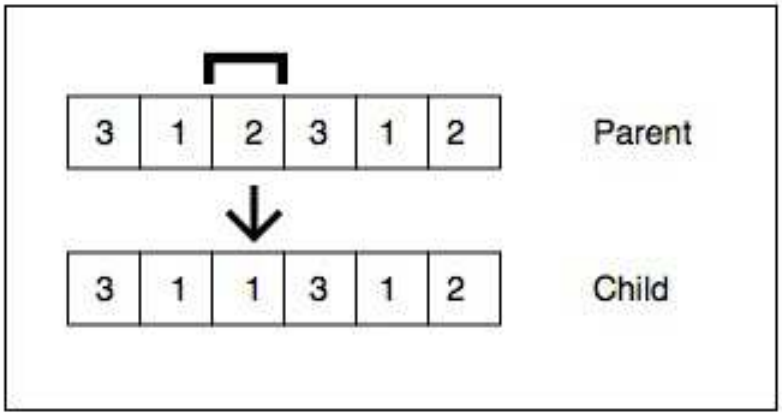

Figure 4.3: Mutation

Chapter 5

Implementation and Experimental Results

This chapter discusses the setting up of cloud environment for implementation of the algorithm. Further this chapter also presents the evaluation work done and the results obtained.

5.1 Cloudsim Description

Cloudsim is a cloud simulator proposed by R. Buyya which is an extension of GridSim [66]. Figure 5.1 models the framework and layered architecture of CloudSim toolkit. CloudSim framework provides virtualized environment including data centers, hosts, virtual machines, cloudlets and also manages different issues in cloud such as provisioning, scheduling. Different policies for allocating host to the VM can be studied in the provisioning layer. The core VM provisioning scheme can be further extended to implement the customized provisioning policy as per the user need.

The main functionalities of CloudSim are as follows:

- Modeling the cloud.
- Modeling the VM allocation.
- Modeling the cloud market.
- Modeling the network behaviour.
- Modeling a federation of cloud.
- Modeling dynamic workloads.
- Modeling data center power consumption.
- Modeling dynamic entities creation.

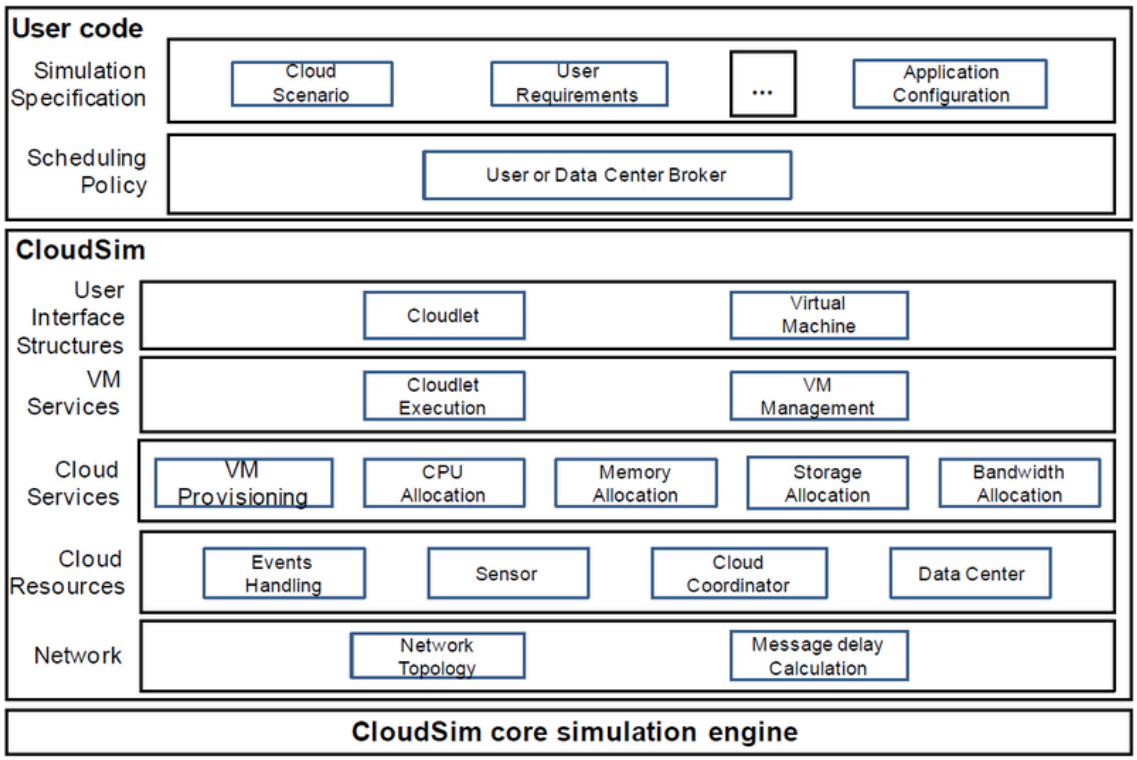

Figure 5.1: CloudSim Architecture [66]

5.2 Implementation Details

The algorithm is implemented on CloudSim kit [66] which provides a platform to perform simulations for cloud environment. In this work, an approach is implemented for task scheduling which is build on GA.

Implementation of the work is described using snapshots.

- **Cloudlet Creation**

  First step is to create cloudlets which is shown in Figure 5.2. In CloudSim tasks are known as cloudlets. Here fixed number of tasks are created each having different length where length of task is randomly assigned.

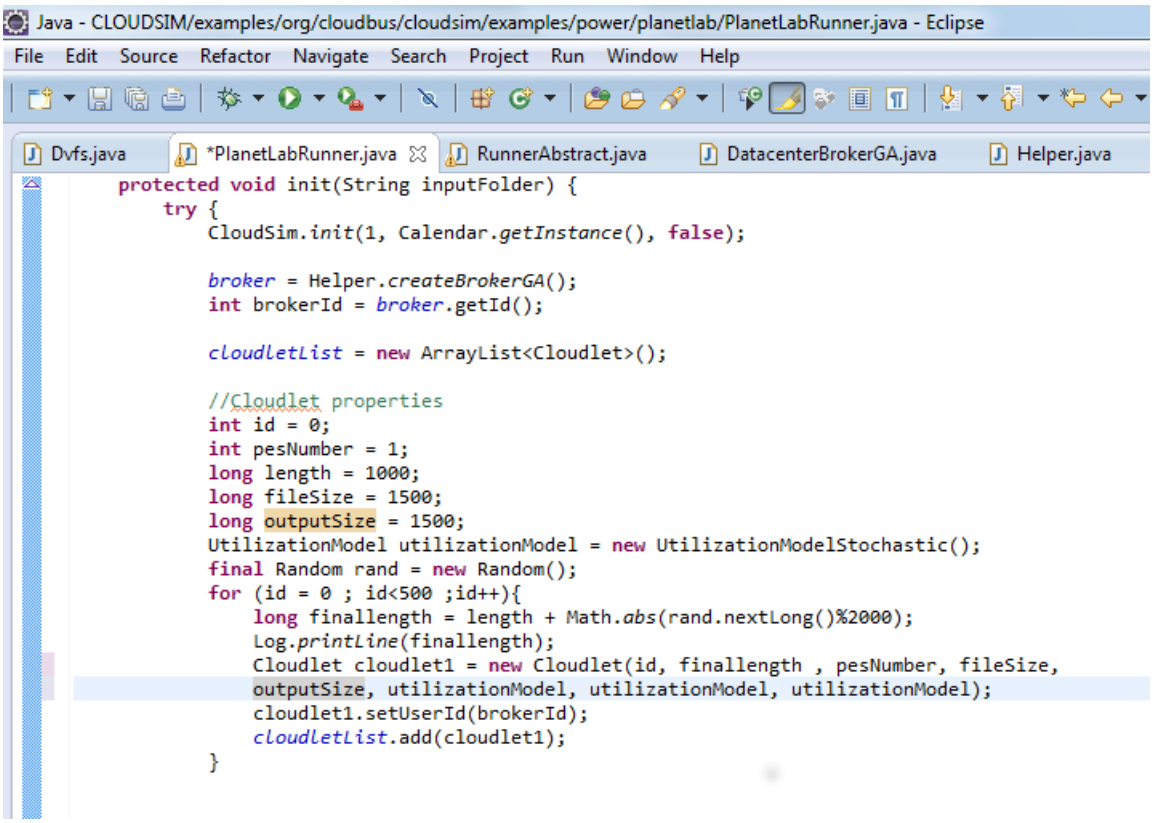

Figure 5.2: Cloudlet Creation

- **VM Creation**

  Next step is to create virtual machines on which tasks are scheduled. VM in CloudSim environment is managed by its host component. Figure 5.3 exhibits how list of virtual machines is created.

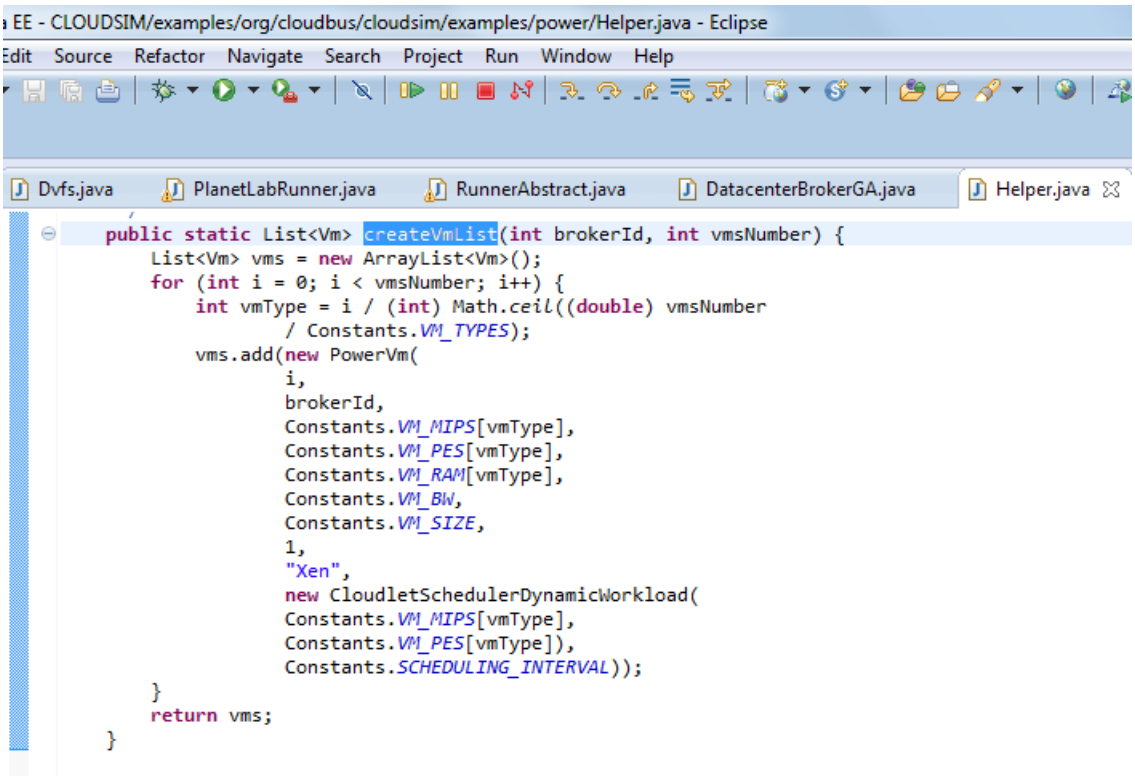

Figure 5.3: VM Creation

- **Host Creation**

  Further list of hosts is created as shown in Figure 5.4. Host is analogous to a physical resource which can be compute or a storage server. It consists of information like memory, storage requirements, processing cores, VM allocation policy and memory provisioning.

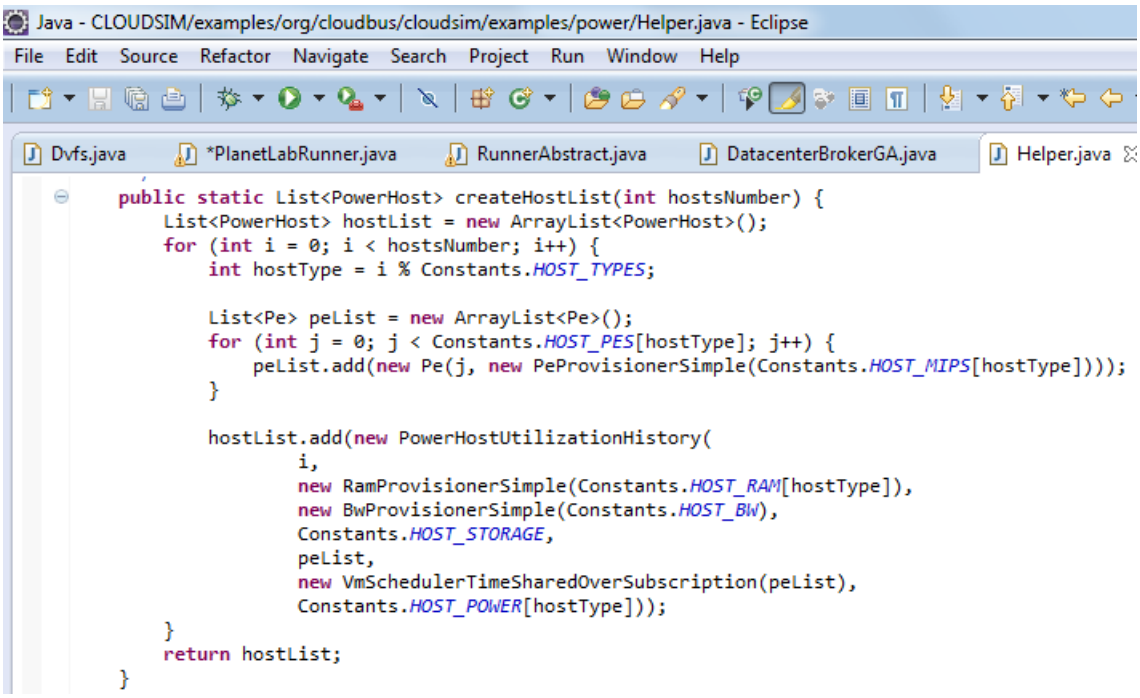

Figure 5.4: Host Creation

- **Datacenter Creation**

  Figure 5.5 depicts the Creation of datacenter. Datacenter models the hardware services and implements policies for allocation of bandwidth, memory and storage to VM and hosts. Datacenter characteristics represent configuration details of the resources.

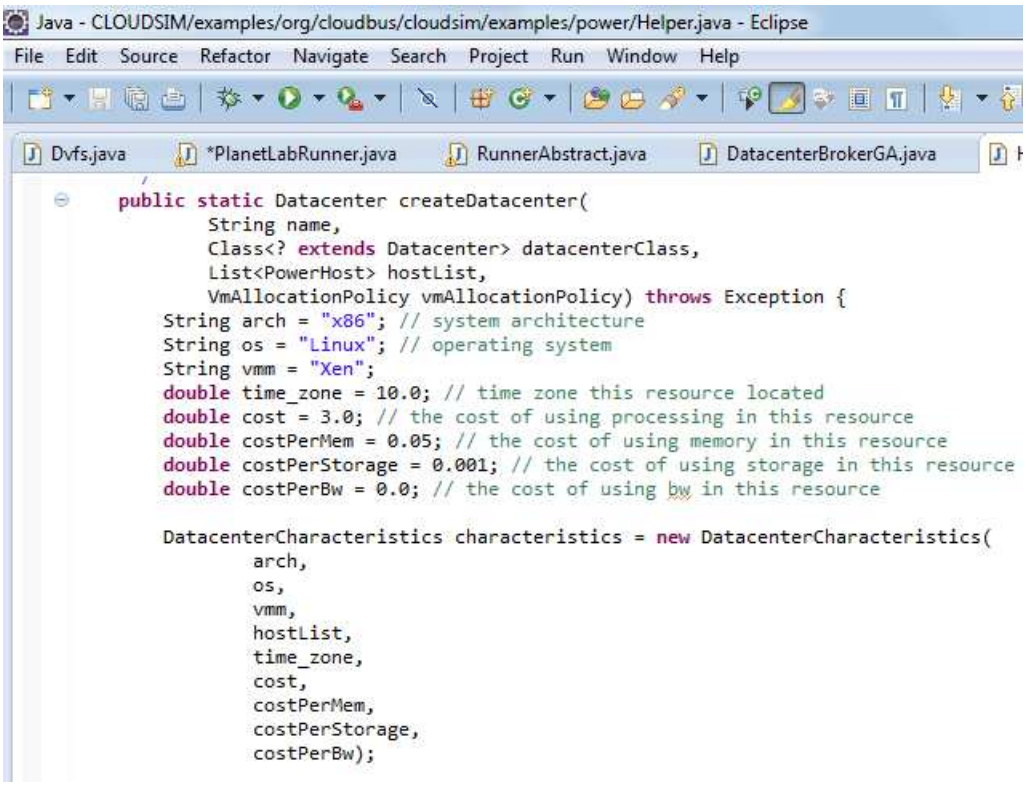

Figure 5.5: Datacenter Creation

- **DatacenterBroker for Genetic Algorithm**

  The default datacenterBroker available in CloudSim is extended to implement the proposed scheduling approach.

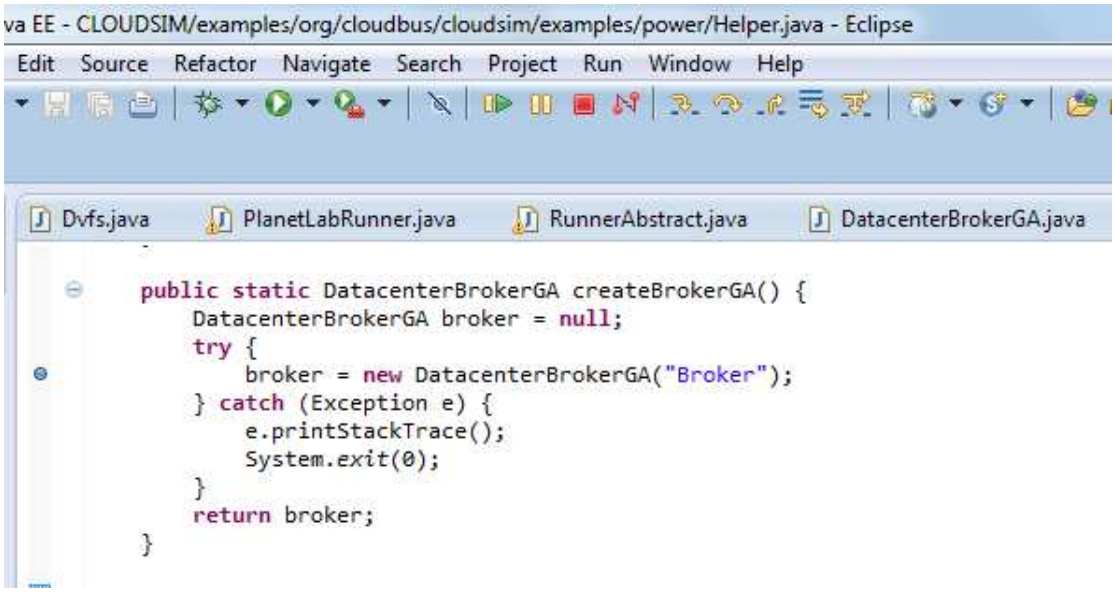

Figure 5.6: DatacenterBroker for Genetic Algorithm

Figure 5.7 describes the implementation of Genetic Algorithm. Initial popu-

lation for genetic algorithm is generated randomly of size 100. Initially VMs are randomly assigned to the cloudlets. Then parents are selected using 4-way tournament selection to apply crossover and mutation. Finally best child is selected and new population is generated. This algorithm returns the list of best VMs assigned to each task.

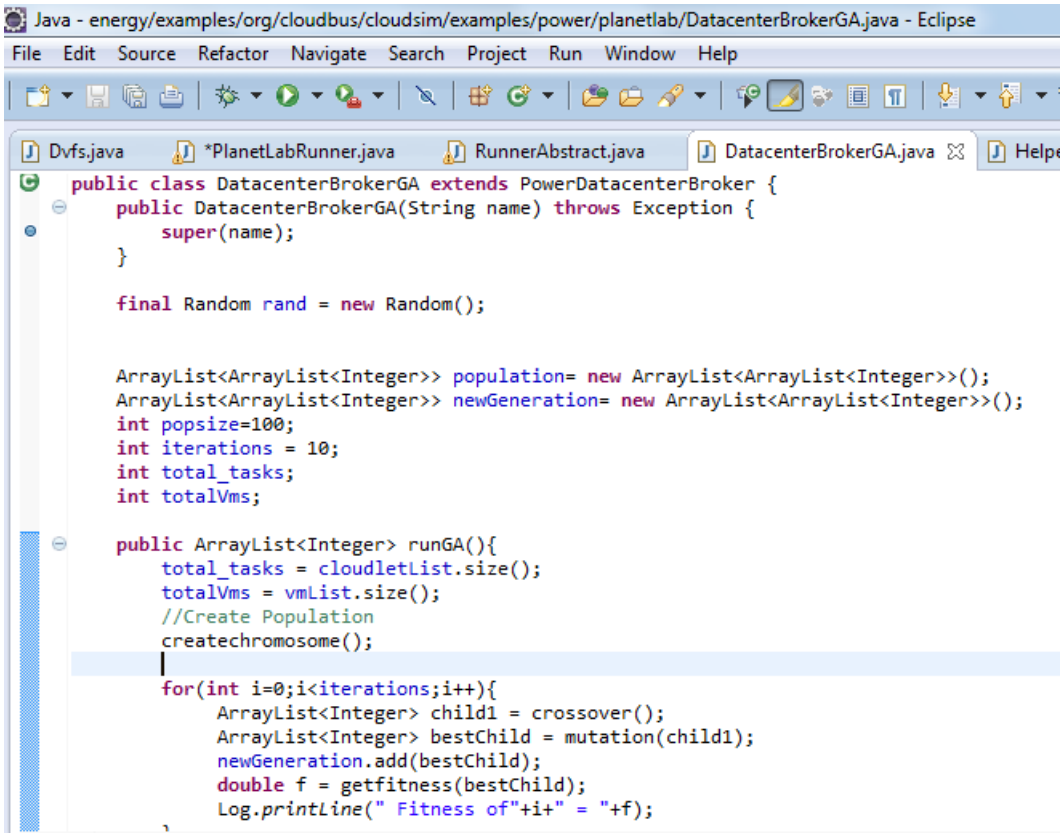

Figure 5.7: Implementation of Genetic Algorithm

Figure 5.8 shows how to submit cloudlets and set VMId for each task which is returned by the implemented genetic algorithm.

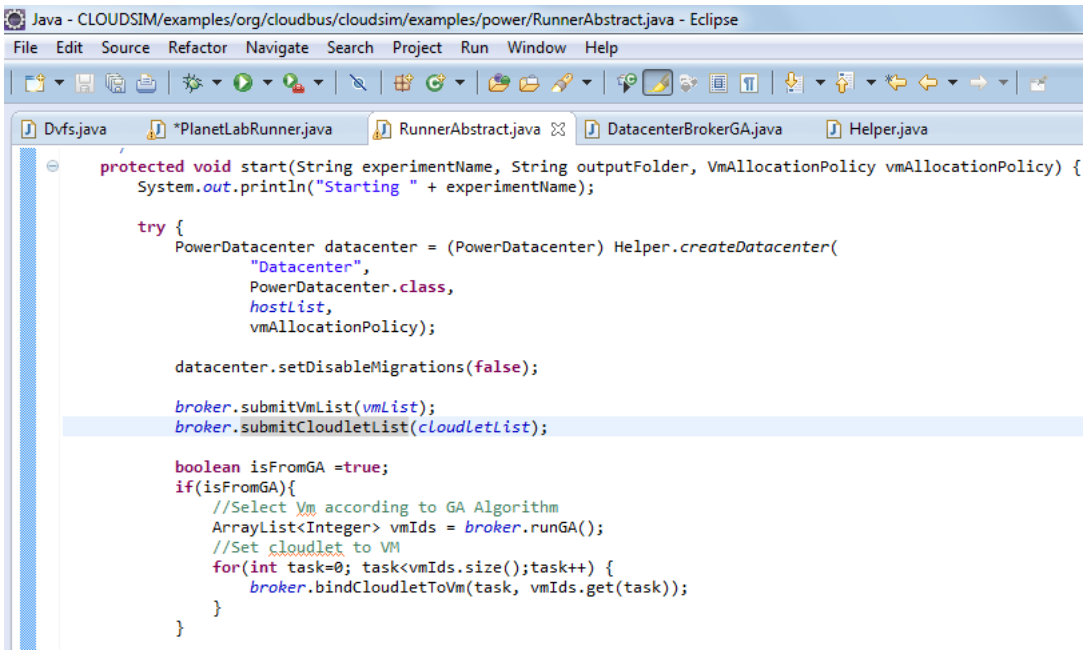

Figure 5.8: Scheduling using Genetic Algorithm

5.3 Evaluation Results

The proposed genetic approach selects the virtual machine according to the utilization. The algorithm returns the list consisting of ids of virtual machine on which task is allocated. The main focus of this scheduling approach is to minimise the energy consumption.

The simulation parameters, hardware requirements and their description is shown in the Table 5.1. Simulation is performed on linux operating system.

Table 5.1: Experiment Parameters

Parameter	Description
Platform	CloudSim
Programming Language	Java
Operating System	Linux
RAM	1GB
Crossover rate	0.6
Mutation rate	0.5
No of iterations	100

Performance analysis of proposed algorithm is done by comparing it with First Come First Serve (FCFS) and Shortest Job First (SJF) algorithm considering energy consumption as the evaluation parameter. One instance of output of these algorithms is shown in the figures below:

- Ouput for FCFS Algorithm

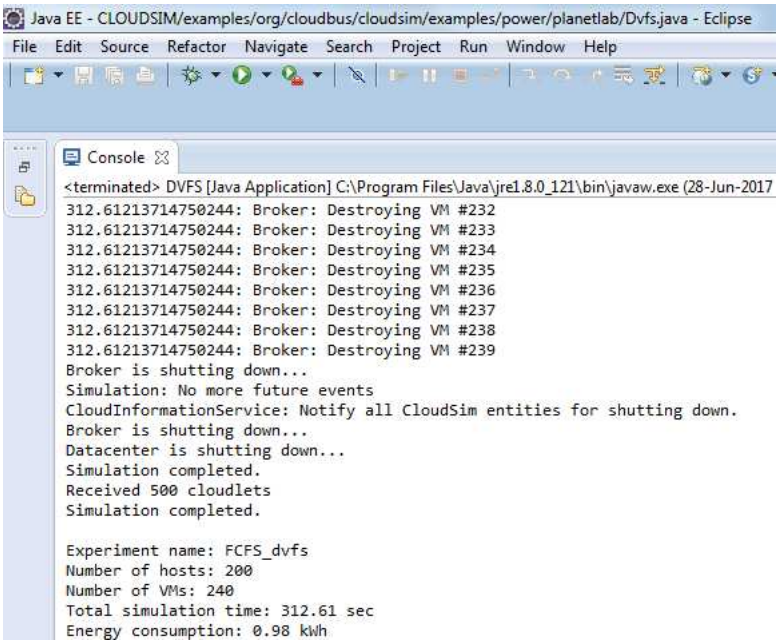

Figure 5.9: FCFS ouput

- Ouput for SJF Algorithm

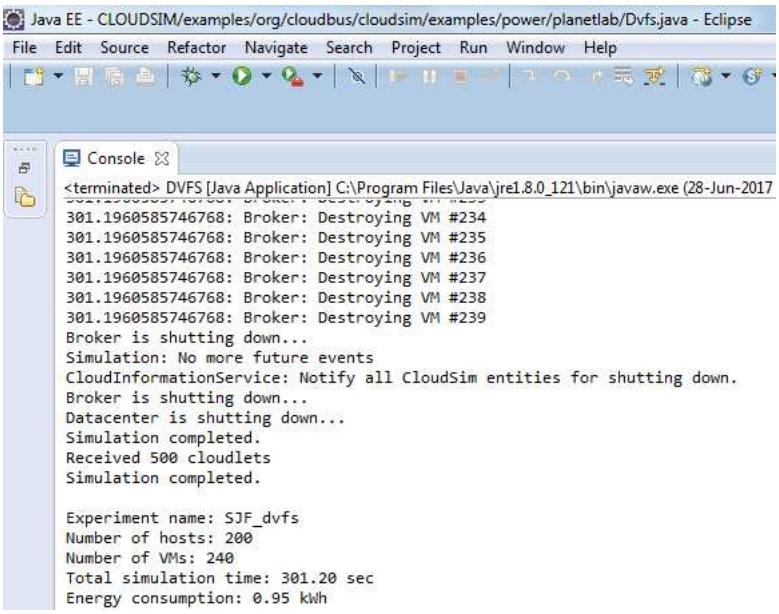

Figure 5.10: SJF ouput

- Ouput for Genetic Algorithm

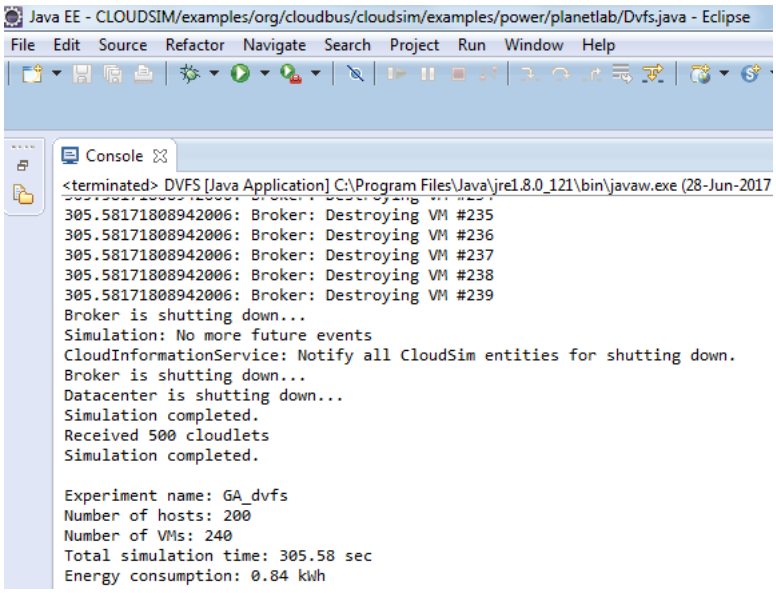

Figure 5.11: GA ouput

Different cases are examined by varying total count of both cloudlets and virtual machines where length of each cloudlet is also different.

- Case I: Number of cloudlets=100, Number of VM=40

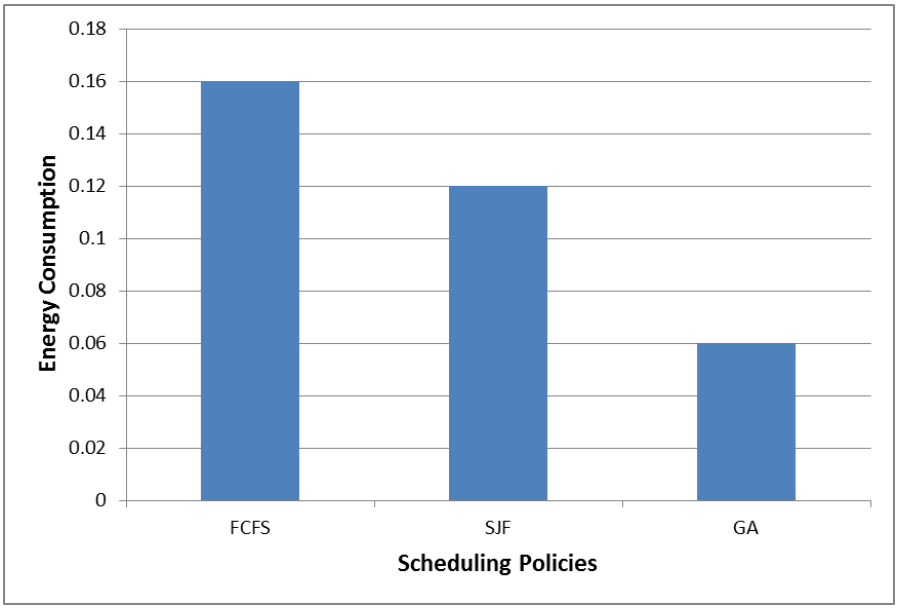

Figure 5.12: Task scheduling with 100 cloudlets and 40 VM

- Case II: Number of cloudlets=200, Number of VM=80

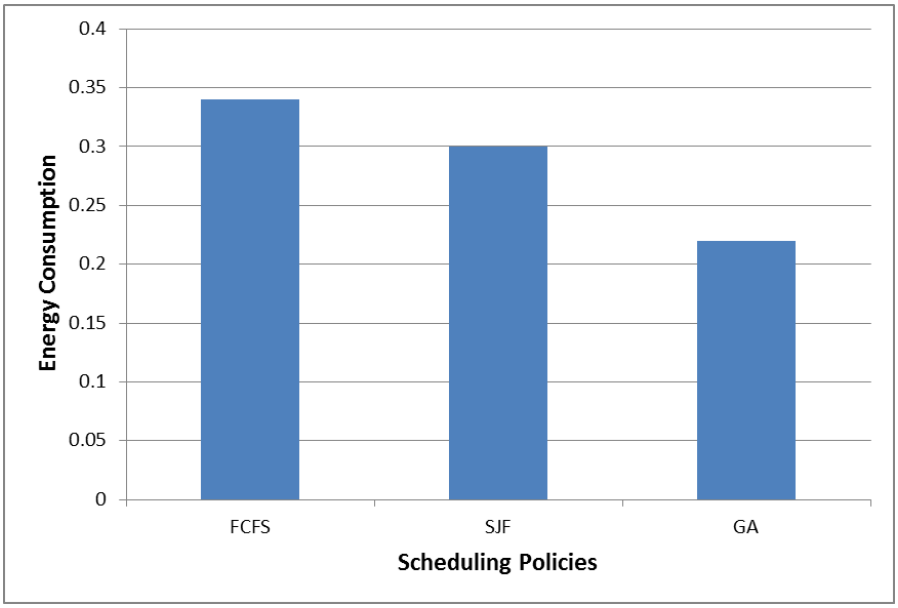

Figure 5.13: Task scheduling with 200 cloudlets and 80 VM

- Case III: Number of cloudlets=300, Number of VM=120

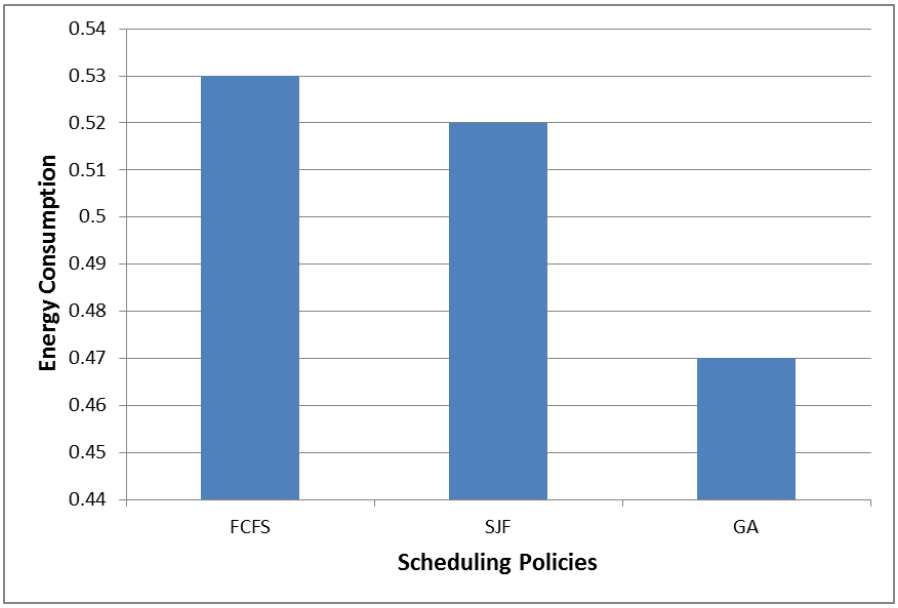

Figure 5.14: Task scheduling with 300 cloudlets and 120 VM

- Case IV: Number of cloudlets=400, Number of VM=180

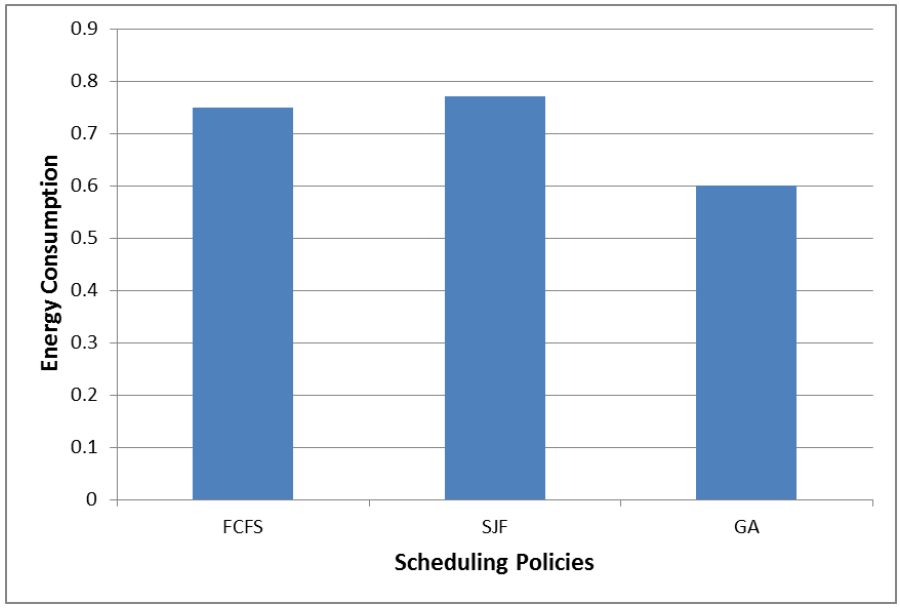

Figure 5.15: Task scheduling with 400 cloudlets and 180 VM

- Case V: Number of cloudlets=500, Number of VM=250

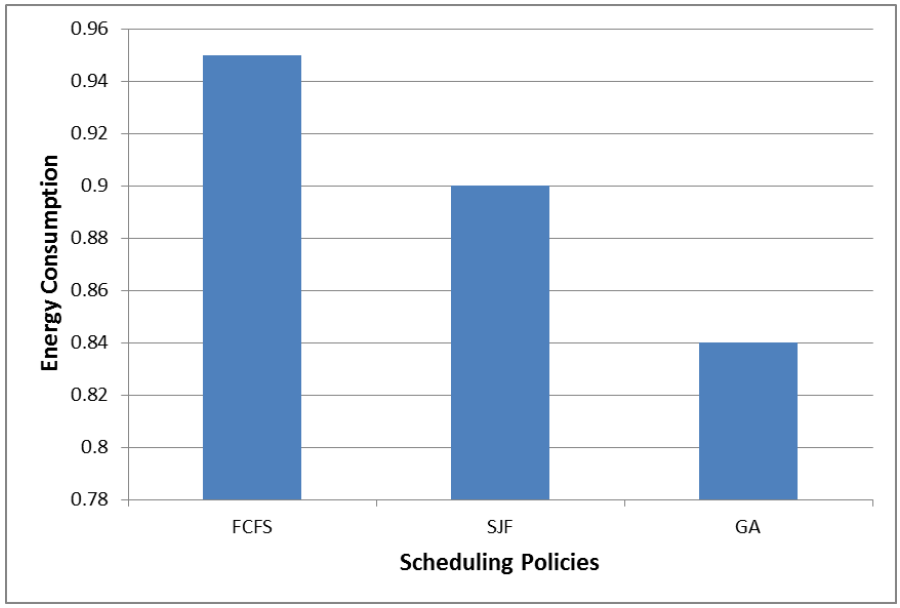

Figure 5.16: Task scheduling with 500 cloudlets and 250 VM

After executing different cases by varying the number of cloudlets and VMs it is concluded that:

- As the number of cloudlets are increased energy consumption also increases.
- Energy consumption in case of SJF algorithm is less than that of FCFS algorithm.

- Scheduling using GA results in least energy consumption as compared to both the algorithms.

Chapter 6

Conclusion and Future Scope

6.1 Conclusion

Cloud computing is the fast growing technology these days. Task scheduling has a major impact on the overall performance of the execution of tasks. Many task scheduling algorithms have been implemented but it is observed that traditional task scheduling algorithms do not give best results. Heuristic and meta-heuristic techniques produce better results for np complete problems and are preferred for task scheduling produce. Among these existing heuristic techniques for cloud computing GA is considered to be a fair algorithm for efficient task scheduling.

6.2 Thesis Contribution

In this research work is done on the issue of energy consumption and an approach is proposed to schedule the tasks based on GA. Virtual machines are selected according to resource utilization and then tasks are mapped on these virtual machines.

Performance of implemented scheduling approach is compared with FCFS and SJF algorithm and experiment results indicated that the proposed scheduling policy outperforms other algorithms and lowers the overall energy.

6.3 Future Scope

In this work, tasks are scheduled on the basis of resource utilization using GA. In future parallel GA can be implemented and workload of each task could also be considered for scheduling. There is a scope to design autonomic computing based self monitoring and self optimizing task scheduling policy considering real time scenarios.

References

[1] D. C. Wyld, *Moving to the cloud: An introduction to cloud computing in government*. IBM Center for the Business of Government, 2009.

[2] S. Zhang, X. Chen, S. Zhang, and X. Huo, "The comparison between cloud computing and grid computing," in *Computer Application and System Modeling (ICCASM), 2010 International Conference on*, vol. 11. IEEE, 2010, pp. V11–72.

[3] G. Pallis, "Cloud computing: the new frontier of internet computing," *IEEE internet computing*, vol. 14, no. 5, pp. 70–73, 2010.

[4] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, and I. Stoica, "A view of cloud computing," *Communications of the ACM*, vol. 53, no. 4, pp. 50–58, 2010.

[5] P. Mell, T. Grance *et al.*, "The nist definition of cloud computing," 2011.

[6] A. Mali, S. Mane, and V. Bhise, "Heuristics for load balancing of vms in cloud," in *3rd Int. Conference on Computational Intelligence and Information Technology–CIIT*, 2013, pp. 115–121.

[7] K. So, "Cloud computing security issues and challenges," *International Journal of Computer Networks*, vol. 3, no. 5, pp. 247–55, 2011.

[8] S. Logo, "Introduction to cloud computing."

[9] M. Eden, "A survey of performance modeling and analysis issues in resource management across x86-based hypervisors in enterprise cloud computing deployments," 2011.

[10] J. Hurwitz, R. Bloor, M. Kaufman, and F. Halper, *Cloud computing for dummies*. John Wiley & Sons, 2010.

[11] A. Singh, "An introduction to virtualization," *kernelthread. com, January*, 2004.

[12] K. Gouda, A. Patro, D. Dwivedi, and N. Bhat, "Virtualization approaches in cloud computing," *International Journal of Computer Trends and Technology (IJCTT)*, vol. 12, no. 4, pp. 161–166, 2014.

[13] I. Education, "Virtualization in education," *IBM Corporation, Whitepaper*, 2007.

[14] Q. Zhang, L. Cheng, and R. Boutaba, "Cloud computing: state-of-the-art and research challenges," *Journal of internet services and applications*, vol. 1, no. 1, pp. 7–18, 2010.

[15] S. Atiewi, S. Yussof, M. Ezanee, and M. Almiani, "A review energy-efficient

task scheduling algorithms in cloud computing," in *Long Island Systems, Applications and Technology Conference (LISAT), 2016 IEEE*. IEEE, 2016, pp. 1–6.

[16] S. Singh and I. Chana, "A survey on resource scheduling in cloud computing: Issues and challenges," *Journal of Grid Computing*, vol. 14, no. 2, pp. 217–264, 2016.

[17] X. Wu, M. Deng, R. Zhang, B. Zeng, and S. Zhou, "A task scheduling algorithm based on qos-driven in cloud computing," *Procedia Computer Science*, vol. 17, pp. 1162–1169, 2013.

[18] D. Kumar and B. Sahoo, "Energy efficient heuristic resource allocation for cloud computing," 2014.

[19] T. A. Henzinger, A. V. Singh, V. Singh, T. Wies, and D. Zufferey, "Static scheduling in clouds," *memory*, vol. 200, no. o1, p. i1.

[20] L. Ma, Y. Lu, F. Zhang, and S. Sun, "Dynamic task scheduling in cloud computing based on greedy strategy," in *International Conference on Trustworthy Computing and Services*. Springer, 2012, pp. 156–162.

[21] L. Guo, S. Zhao, S. Shen, and C. Jiang, "Task scheduling optimization in cloud computing based on heuristic algorithm." *JNW*, vol. 7, no. 3, pp. 547–553, 2012.

[22] A. Bala and I. Chana, "A survey of various workflow scheduling algorithms in cloud environment," in *2nd National Conference on Information and Communication Technology (NCICT)*. sn, 2011, pp. 26–30.

[23] D. Agarwal and S. Jain, "Efficient optimal algorithm of task scheduling in cloud computing environment," *arXiv preprint arXiv:1404.2076*, 2014.

[24] P. Salot, "A survey of various scheduling algorithm in cloud computing environment," *International Journal of Research in Engineering and Technology*, vol. 2, no. 2, pp. 131–135, 2013.

[25] T. D. Braun, H. J. Siegel, N. Beck, L. L. Bölöni, M. Maheswaran, A. I. Reuther, J. P. Robertson, M. D. Theys, B. Yao, and D. Hensgen, "A comparison of eleven static heuristics for mapping a class of independent tasks onto heterogeneous distributed computing systems," *Journal of Parallel and Distributed computing*, vol. 61, no. 6, pp. 810–837, 2001.

[26] H. Izakian, A. Abraham, and V. Snasel, "Comparison of heuristics for scheduling independent tasks on heterogeneous distributed environments," in *Computational Sciences and Optimization, 2009. CSO 2009. International Joint Conference on*, vol. 1. IEEE, 2009, pp. 8–12.

[27] S. H. H. Madni, M. S. A. Latiff, M. Abdullahi, and M. J. Usman, "Performance comparison of heuristic algorithms for task scheduling in iaas cloud computing environment," *PloS one*, vol. 12, no. 5, p. e0176321, 2017.

[28] O. Elzeki, M. Reshad, and M. Elsoud, "Improved max-min algorithm in cloud computing," *International Journal of Computer Applications*, vol. 50, no. 12, 2012.

[29] U. Bhoi and P. N. Ramanuj, "Enhanced max-min task scheduling algorithm in cloud computing," *International Journal of Application or Innovation in Engineering and Management*, vol. 2, no. 4, pp. 259–64, 2013.

[30] G. Liu, J. Li, and J. Xu, "An improved min-min algorithm in cloud computing," in *Proceedings of the 2012 International Conference of Modern Computer Science and Applications*. Springer, 2013, pp. 47–52.

[31] H. Chen, F. Wang, N. Helian, and G. Akanmu, "User-priority guided min-min scheduling algorithm for load balancing in cloud computing," in *Parallel Computing Technologies (PARCOMPTECH), 2013 National Conference on*. IEEE, 2013, pp. 1–8.

[32] G. Gupta, V. K. Kumawat, P. Laxmi, D. Singh, V. Jain, and R. Singh, "A simulation of priority based earliest deadline first scheduling for cloud computing system," in *Networks & Soft Computing (ICNSC), 2014 First International Conference on*. IEEE, 2014, pp. 35–39.

[33] C.-W. Tsai and J. J. Rodrigues, "Metaheuristic scheduling for cloud: A survey," *IEEE Systems Journal*, vol. 8, no. 1, pp. 279–291, 2014.

[34] M. Kalra and S. Singh, "A review of metaheuristic scheduling techniques in cloud computing," *Egyptian informatics journal*, vol. 16, no. 3, pp. 275–295, 2015.

[35] X. Liu and J. Liu, "A task scheduling based on simulated annealing algorithm in cloud computing," *International Journal of Hybrid Information Technology*, vol. 9, no. 6, pp. 403–412, 2016.

[36] S. C. Porto and C. C. Ribeiro, "A tabu search approach to task scheduling on heterogeneous processors under precedence constraints," *International Journal of high speed computing*, vol. 7, no. 01, pp. 45–71, 1995.

[37] C. Zhao, S. Zhang, Q. Liu, J. Xie, and J. Hu, "Independent tasks scheduling based on genetic algorithm in cloud computing," in *Wireless Communications, Networking and Mobile Computing, 2009. WiCom'09. 5th International Conference on*. IEEE, 2009, pp. 1–4.

[38] M. A. Tawfeek, A. El-Sisi, A. E. Keshk, and F. A. Torkey, "Cloud task scheduling based on ant colony optimization," in *Computer Engineering & Systems (ICCES), 2013 8th International Conference on*. IEEE, 2013, pp. 64–69.

[39] V. V. Kumar and K. Dinesh, "Job scheduling using fuzzy neural network algorithm in cloud environment," *Bonfring International Journal of Man Machine Interface*, vol. 2, no. 1, p. 1, 2012.

[40] B. Xu, C. Zhao, E. Hu, and B. Hu, "Job scheduling algorithm based on berger model in cloud environment," *Advances in Engineering Software*, vol. 42, no. 7, pp. 419–425, 2011.

[41] W. Wang, G. Zeng, D. Tang, and J. Yao, "Cloud-dls: Dynamic trusted scheduling for cloud computing," *Expert Systems with Applications*, vol. 39, no. 3, pp. 2321–2329, 2012.

[42] S. Su, J. Li, Q. Huang, X. Huang, K. Shuang, and J. Wang, "Cost-efficient task scheduling for executing large programs in the cloud," *Parallel Computing*, vol. 39, no. 4, pp. 177–188, 2013.

[43] S. Abrishami, M. Naghibzadeh, and D. H. Epema, "Deadline-constrained workflow scheduling algorithms for infrastructure as a service clouds," *Future Generation Computer Systems*, vol. 29, no. 1, pp. 158–169, 2013.

[44] J. L. Lucas-Simarro, R. Moreno-Vozmediano, R. S. Montero, and I. M. Llorente, "Scheduling strategies for optimal service deployment across multiple clouds," *Future Generation Computer Systems*, vol. 29, no. 6, pp. 1431–1441, 2013.

[45] S. Parsa and R. Entezari-Maleki, "Rasa: A new task scheduling algorithm in grid environment," *World Applied sciences journal*, vol. 7, no. Special issue of Computer & IT, pp. 152–160, 2009.

[46] A. G. Delavar, M. Javanmard, M. B. Shabestari, and M. K. Talebi, "Rsdc (reliable scheduling distributed in cloud computing)," *International Journal of Computer Science, Engineering and Applications*, vol. 2, no. 3, p. 1, 2012.

[47] D. M. Dakshayini and D. H. Guruprasad, "An optimal model for priority based service scheduling policy for cloud computing environment," *International Journal of Computer Applications*, vol. 32, no. 9, pp. 23–29, 2011.

[48] S. K. Garg, C. S. Yeo, A. Anandasivam, and R. Buyya, "Energy-efficient scheduling of hpc applications in cloud computing environments," *arXiv preprint arXiv:0909.1146*, 2009.

[49] K. H. Kim, A. Beloglazov, and R. Buyya, "Power-aware provisioning of cloud resources for real-time services," in *Proceedings of the 7th International Workshop on Middleware for Grids, Clouds and e-Science*. ACM, 2009, p. 1.

[50] G. Gharooni-fard, F. Moein-darbari, H. Deldari, and A. Morvaridi, "Scheduling of scientific workflows using a chaos-genetic algorithm," *Procedia Computer Science*, vol. 1, no. 1, pp. 1445–1454, 2010.

[51] T. Wang, Z. Liu, Y. Chen, Y. Xu, and X. Dai, "Load balancing task scheduling based on genetic algorithm in cloud computing," in *Dependable, Autonomic and Secure Computing (DASC), 2014 IEEE 12th International Conference on*. IEEE, 2014, pp. 146–152.

[52] Y. Ge and G. Wei, "Ga-based task scheduler for the cloud computing sys-

tems," in *Web Information Systems and Mining (WISM), 2010 International Conference on*, vol. 2. IEEE, 2010, pp. 181–186.

[53] R. Buyya, A. Beloglazov, and J. Abawajy, "Energy-efficient management of data center resources for cloud computing: a vision, architectural elements, and open challenges," *arXiv preprint arXiv:1006.0308*, 2010.

[54] I. Sari, "Optimizing energy consumption in clouds by using genetic algorithm."

[55] S. Selvarani and G. S. Sadhasivam, "Improved cost-based algorithm for task scheduling in cloud computing," in *Computational intelligence and computing research (iccic), 2010 ieee international conference on*. IEEE, 2010, pp. 1–5.

[56] X. Zhu, C. He, K. Li, and X. Qin, "Adaptive energy-efficient scheduling for real-time tasks on dvs-enabled heterogeneous clusters," *Journal of parallel and distributed computing*, vol. 72, no. 6, pp. 751–763, 2012.

[57] L. Sawalha and R. D. Barnes, "Energy-efficient phase-aware scheduling for heterogeneous multicore processors," in *Green Technologies Conference, 2012 IEEE*. IEEE, 2012, pp. 1–6.

[58] J. Liu, X.-G. Luo, X.-M. Zhang, F. Zhang, and B.-N. Li, "Job scheduling model for cloud computing based on multi-objective genetic algorithm," *IJCSI International Journal of Computer Science Issues*, vol. 10, no. 1, pp. 134–139, 2013.

[59] M. Mezmaz, N. Melab, Y. Kessaci, Y. C. Lee, E.-G. Talbi, A. Y. Zomaya, and D. Tuyttens, "A parallel bi-objective hybrid metaheuristic for energy-aware scheduling for cloud computing systems," *Journal of Parallel and Distributed Computing*, vol. 71, no. 11, pp. 1497–1508, 2011.

[60] N. Quang-Hung, P. D. Nien, N. H. Nam, N. H. Tuong, and N. Thoai, "A genetic algorithm for power-aware virtual machine allocation in private cloud," in *Information and Communication Technology-EurAsia Conference*. Springer, 2013, pp. 183–191.

[61] E. Feller, L. Rilling, and C. Morin, "Energy-aware ant colony based workload placement in clouds," in *Proceedings of the 2011 IEEE/ACM 12th International Conference on Grid Computing*. IEEE Computer Society, 2011, pp. 26–33.

[62] M. Qiu, Z. Ming, J. Li, K. Gai, and Z. Zong, "Phase-change memory optimization for green cloud with genetic algorithm," *IEEE Transactions on Computers*, vol. 64, no. 12, pp. 3528–3540, 2015.

[63] F. Tao, Y. Feng, L. Zhang, and T. W. Liao, "Clps-ga: A case library and pareto solution-based hybrid genetic algorithm for energy-aware cloud service scheduling," *Applied Soft Computing*, vol. 19, pp. 264–279, 2014.

[64] A. Y. Zomaya and Y.-H. Teh, "Observations on using genetic algorithms

for dynamic load-balancing," *IEEE transactions on parallel and distributed systems*, vol. 12, no. 9, pp. 899–911, 2001.

[65] G. L. Valentini, S. U. Khan, and P. Bouvry, "Energy-efficient resource utilization in cloud computing," *Large Scale Network-centric Computing Systems, John Wiley & Sons, Hoboken, NJ, USA*, 2013.

[66] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. De Rose, and R. Buyya, "Cloudsim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," *Software: Practice and experience*, vol. 41, no. 1, pp. 23–50, 2011.

List of Publications

Communicated

[1] Paridhi Naithani, Dr. Rajesh Kumar, *"Genetic Algorithm Based Scheduling To Reduce Energy Consumption In Cloud*, 4th IEEE International Conference on Signal Processing, Computing and Control (ISPCC 2017), Solan, 21-23 September 2017.